sidr B. Dickson
Internet-Draft Brian Dickson
Expires: September 6, 2012 March 5, 2012

Route Leaks -- Definitions
draft-dickson-sidr-route-leak-def-01

Abstract

The Border Gateway Protocol, version 4, (BGP4) provides the means to advertise reachability for IP prefixes. This reachability information is propagated in a peer-to-peer topology. Sometimes routes are announced to peers for which the local peering policy does not permit. And sometimes routes are propagated indiscriminantly, once they have been accepted.

This document considers the situations that can lead to routes being leaked, and tries to find acceptable definitions for describing these scenarios.

The purpose of these definitions is to facilitate analysis of what a route leak is, and what the scope of the problem space for route leaks is.

This, in turn, is intended to inform a requirements document for detection of (and prevention of) route leaks. And finally, the definitions and requirements are intended to allow proposed solutions which meet these criteria, and to facilitate evaluation of proposed solutions.

The fundamental objective is to "solve the route leaks problem".

Author’s Note

Intended Status: Informational.

Status of this Memo

This Internet-Draft will expire on September 6, 2012.

Copyright Notice

Table of Contents

1. Introduction

1.1. Rationale

A route-leak occurs when a prefix is originated by one party, propagated by other parties, and received by the observer, where the path used was not intentional end-to-end. It is a leak if the receiver did not want the route, from a generic policy perspective. It does not matter which party caused the situation - a leak is in the eye of the receiver. By their nature - unintentional, unwanted, and harmful, route leaks are bad.

By first establishing a more precise definition of route leak, the intent is to find requirements for mechanisms for stopping route leaks, and then finding solutions that meet those requirements.

1.2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.3. Terminology

The reader is assumed to be familiar with BGP version 4, both from a protocol perspective and from an operational perspective. BGP4 is defined in [RFC1771], and updated or enhanced by a variety of other RFCs.

The following terminology is used throughout this document:

Route (or synonomously, prefix): an NLRI in BGP, including all its attributes.

Neighbor (or "peer", not capitalized): A toplogically adjacent Autonomous System, with whom routes are exchanged.

Link: A BGP connection to a Neighbor. A Neighbor may be reached via one or more links.

Link Classification: The "intent" of a given BGP peering session, which addresses only the categories of route announced and accepted, and which is further modified by Local Policy.

Local Policy: The set of rules, as applied on a single Neighbor Link, on which routes are announced, which routes are accepted, and what attributes are changed to affect choice of BGP Best Path per prefix.

Path: Also known as AS Path, the sequence of ASNs through which a route has passed from Originator to recipient.

Hijacked Route: A route which has been originated by a party other than the owner of the prefix. This could be via a forged ASN, or from another ASN.

Validated Origination: a route whose origination has been validated via cryptographic means, using an ROA.

Link Classifications: a Link may be classified as:

- Customer
- Transit
- Peer
- Mutual Transit

2. Scope Limitations

The following issues are not in the scope of route leaks. Each item in the list includes the rationale for excluding it.

Hijacked Routes:
Origin Validation already addresses the issue of Hijacked Routes. By limiting Route Leak efforts to Validated Routes, we are able to presume the origin is correct.

Violations of Local Policy:
Issues between adjacent ASNs which do not propogate any further, or which do not violate the Link Classification.

Other-ASN Relationship:
The "correctness" of a given prefix received over a Link, is determined only by the Link Classifications of each Link in the Path. The existence of other Links, to Neighbors with ASNs on a given Path (which may have differing Link Classifications), is a classic "apples to oranges" comparision. It is incorrect to compare ASNs outside the context of the AS path, so we exclude those comparisons from this work.

Essentially, the only elements being considered are the Path, and Link Classifications at each hop in the Path.

3. Route Leak Definitions

Route Leak Initiation: A Route announced over a Link by a Neighbor which does not match the Link Classification, where the Neighbor is either the Originator, or had received the Route where the Neighbor's Link Classification matched the Route that the Neighbor received. In lay terms, this means that the Neighbor is the party that caused the route leak, by announcing a route contrary to the Link Classification (and consequently also violated the Local Policy).

Route Leak Propagation: A Route announced over a Link by a Neighbor, where the Neighbor received the Route as either a Route Leak Initiation, or a Route Leak Propagation. A Route Leak Propagation may appear to match the Link Classification, since the Path appears similar to non-leaked routes for the first two ASNs in the Path.

3.1. Peer Links and Routes

A Peer Classification is a Link over which the two parties send only their respective Customer Routes (and their Customer's Routes, and so on).

A Link which is classified as a Peer, will see us as a Peer Classification as well. The relationship is symmetric in nature.

3.2. Customer Links and Routes

A Customer Link Classification: The Customer sends us only their own Routes, and the Customer's Customer's Routes (and Customer^Nth Routes). The Customer relationship is transitive.

A Transit Link Classification: The Transit provider sends all Routes. This include the Transit Provider's Customers, the Transit Provider's Peers, and if there are any, the Transit Provider's Transit Provider's Routes. The Transit Provider relationship is also transitive.

Transit and Customer are the opposite ends of the same Link, by definition.

The Customer Classification is a superset of the actual Local Policy of a specific Customer. This means that while a Customer Classification means "we send all routes", the actual Local Policy for a specific Customer might differ, and the Customer might only receive some Routes, or none at all. Similarly, the Classification means that we are prepared to accept the Customer's own Routes, as well as those of the Customer's Customers. However, the Local Policy might be to accept only a specific subset of the Customer's Routes.

3.2.1. Customer’s Customer

It is important to define when a Route is a Customer’s Customer Route.

A Customer’s Customer Route: the Path to be from the Customer’s Customer, to the Customer, to us. Similarly, Customer^Nth Paths must proceed directly from Customer^N to Customer^(N-1) to Customer to us. It is not sufficient for the Origin of the Route to be the ASN of a Customer’s Customer. Each Link must be a Customer Classification, or Mutual Transit, which is a superset of Customer.

3.3. Mutual Transit

A Mutual Transit Classification is a Link where the two parties agree to provide full routes, and to advertise each others’ customers routes the same as they would advertise their own customers’ routes. Semantically, this behaves the same as having two Links where one is Transit and the other is Customer.

3.4. Non-Initiation Links

To help identify the exact conditions where a Route Leak Initiation can occur, it is helpful to exclude Link Classifications where it is not possible to cause a Route Leak Initiation.

A Transit Classification, by definition, can receive all routes.

Thus, a Transit Classification Link cannot be the source of a Route Leak Initiation.

By the same logic, a Mutual Transit Classification cannot be the source of a Route Leak Initiation.

This leads to a more precise definition of a Route Leak Initiation.

3.5. Route Leak Initiation

Route Leak Initiation: Non-Customer Route received over a Peer or Customer Link.

3.6. Route Leak

Route Leak: any Route where, somewhere in the Path, a Non-Customer Route was received over a Peer or Customer Link. (This is synomous with "was sent over a Peer or Transit Link".)

4. Security Considerations

None per se.

5. IANA Considerations

This document contains no IANA-specific material.

6. Acknowledgements

To be added later.

7. References

7.1. Normative References

[RFC1773] Traina, P., "Experience with the BGP-4 protocol", RFC 1773, March 1995.

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, August 1996.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

7.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author's Address

Brian Dickson
Brian Dickson
703 Palmer Drive,
Herndon, VA 20170
USA

Email: brian.peter.dickson@gmail.com

sidr
Internet-Draft
Expires: April 26, 2013

B. Dickson
Brian Dickson
October 23, 2012

Route Leaks -- Definitions
draft-dickson-sidr-route-leak-def-03

Abstract

The Border Gateway Protocol, version 4, (BGP4) provides the means to advertise reachability for IP prefixes. This reachability information is propagated in a peer-to-peer topology. Routes may be announced to neighbors, contrary to the receiver’s local peering policy. If that occurs, those routes may then be propagated indiscriminantly, once they have been accepted.

This document considers the situations that can lead to routes being leaked, and tries to find acceptable definitions for describing these scenarios.

The purpose of these definitions is to facilitate analysis of what a route leak is, and what the scope of the problem space for route leaks is.

This, in turn, is intended to inform a requirements document for detection of (and prevention of) route leaks. And finally, the definitions and requirements are intended to allow proposed solutions which meet these criteria, and to facilitate evaluation of proposed solutions.

The ultimate goal is to "solve the route leaks problem".

Author’s Note

Intended Status: Informational.

Status of this Memo

This Internet-Draft will expire on April 26, 2013.

Copyright Notice

Table of Contents

1. Introduction

1.1. Assumptions

Much of this document assumes the observer has total knowledge of the state of everything in the hypothetical examples presented.

It is understood that participants in the real world routing scenarios will not have that knowledge.

The purpose of presuming that total knowledge here, is to illustrate how little is needed to identify leaked routes.

In particular, it is hoped that this leads to a correspondingly simple set of definitions with useful real-world meaning.

1.2. Rationale

Generally speaking, a route-leak occurs when a route goes somewhere it should not. In other words, that somewhere along the path, a route was sent that somehow violated the implicit or explicit policy between two neighbors, without being blocked by the recipient. Route leaks cause harm, in a variety of ways. They expose traffic to Man-In-The-Middle (MITM) attacks. They may result in traffic congestion, latency, or even black-holing of traffic.

It is a leak if any receiver in the propogation path did not want the route, from a generic policy perspective. It does not matter which party caused the situation - a leaks are in the eye of the receivers. By their nature - unintentional, unwanted, and harmful - route leaks are bad.

By first establishing a more precise definition of route leak, the intent is to find requirements for mechanisms for stopping route leaks, and then finding solutions that meet those requirements.

1.3. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.4. Terminology

The reader is assumed to be familiar with BGP version 4, both from a protocol perspective and from an operational perspective. BGP4 is defined in [RFC1771], and updated or enhanced by a variety of other RFCs.

The following additional terminology is used throughout this document:

Route (or synonomously, prefix): an NLRI in BGP, including all its attributes. (This term subject to change by GROW.)

Neighbor (or "peer", not capitalized): A toplogically adjacent Autonomous System, with whom routes are exchanged.

Link: A BGP connection to a Neighbor. A Neighbor may be reached via one or more links, where each link may have a different classification, and/or local policy.

Link Classification: The "intent" of a given BGP peering session, which addresses only the categories of route announced and accepted, and which is further modified by Local Policy.

Local Policy: The set of rules, as applied on a single Neighbor Link, specifying which routes are announced, which routes are accepted, and what attributes are changed to affect choice of BGP Best Path per prefix.

Path: Also known as AS_PATH (or optionally AS4_PATH), the sequence of ASNs through which a route has passed from Originator to recipient.

Hijacked Route: A route which has been originated by a party other than the owner of the prefix. This could be via a forged ASN, or from another ASN.

Validated Origination: a route whose origination has been validated, e.g. via cryptographic means, such as using an ROA.

2. Scope Limitations

The following issues are not in the scope of route leaks. Each item in the list includes the rationale for excluding it.

- Hijacked Routes - Origin Validation (proposed work in the SIDR WG) addresses the issue of Hijacked Routes. By limiting Route Leak efforts to Validated Routes, we are able to presume the origin is correct, and narrow the scope.
- Violations of Local Policy - issues between adjacent ASNs which do not propogate any further, or which do not violate the Link Classification.
- Other-ASN Relationship - The "correctness" of a given prefix received over a Link, is determined only by the Link Classifications of each Link in the Path. The existence of other Links, to Neighbors with ASNs on a given Path (which may have

differing Link Classifications), is a classic "apples to oranges" comparision. It is incorrect to compare ASNs outside the context of the AS path, so we exclude those comparisons from this work. Essentially, the only elements being considered are the Path, and Link Classifications at each hop in the Path.

3. Route Leak Definitions

Route Leak Initiation: A Route announced over a Link by a Neighbor, which does not match the Link Classification, where one of the following is true:

- the Neighbor is the Originator
- the Neighbor received the Route, where the received Route was not a Route Leak

In lay terms, this means that the Neighbor is the party that caused the route leak, by announcing a route contrary to the Link Classification (and consequently also violated the Local Policy).

Route Leak Propagation: A Route announced over a Link by a Neighbor, where the Route that the Neighbor received was either a Route Leak Initiation, or a Route Leak Propagation.

Once a Route has become a Route Leak Initiation, any further announcement of that Route is a Route Leak Propagation.

NB: A Route Leak Propagation may appear to match the Link Classification, since the Path appears similar to non-leaked routes for the first two ASNs in the Path.

Link Classifications: a Link may be classified as:

- Customer
- Transit
- Peer
- Special (which includes Mutual Transit, Sibling, and other non-trivial arrangements)

Special (e.g. Mutual Transit): a Link where the two parties agree to provide full routes, and to advertise each others’ customers routes the same as they would advertise their own customers’ routes. Semantically, this behaves the same as having two parallel Links between the same two Neighbors, where one Link Policy is Transit and the other Link Policy is Customer. Recall, Link Classification is the superset of Local Policy - the term "full routes" here means simply that any route in addition to customers’ routes, is permitted.

4. Peer Links and Routes

A Peer Classification is a Link over which the two parties send ONLY their respective Customer Routes (and their Customer’s Routes, and so on).

A Link which is classified as a Peer, will see us as a Peer Classification as well. The relationship is symmetric in nature.

5. Customer Links and Routes

A Customer Link Classification: The Customer sends us only their own (locally originated) Routes, and the Customer’s Customer’s Routes (and Customer^Nth Routes). The Customer relationship is transitive.

A Transit Link Classification: The Transit provider sends all Routes. This include the Transit Provider’s Customers, the Transit Provider’s Peers, and if there are any, the Transit Provider’s Transit Provider’s Routes. The Transit Provider relationship is also transitive.

Transit and Customer are the opposite ends of the same Link, by definition.

The Transit Link Classification is a superset of the actual Local Policy of a specific Customer. This means that while a Transit Link Classification means "we send all routes", the actual Local Policy for a specific Customer might differ, and the Customer might only receive some Routes, or none at all. Similarly, the Classification means that we are prepared to accept the Customer’s own Routes, as well as those of the Customer’s Customers. However, the Local Policy might be to accept only a specific subset of the Customer’s Routes.

5.1. Customer’s Customer

It is important to define when a Route is a Customer’s Customer Route.

A Customer’s Customer Route: the Path to be from the Customer’s Customer, to the Customer, to us. Similarly, Customer^Nth Paths must proceed directly from Customer^N to Customer^(N-1) to Customer to us. It is not sufficient for the Origin of the Route to be the ASN of a Customer’s Customer. Each Link must be a Customer Classification (or Special, e.g. Mutual Transit, which is a superset of Customer).

In particular, if the Path were to include any Link which were not a

Customer Link, the Route would NOT be a Customer^N.

NB: It is sufficient that the Customer’s Customer relationship is declared. The "Customer" relationship, in the context of route leaks, is restrictive. Erroneous or inadvertent classification as Customer cannot result in a route leak.

6. Non-Leak-Initiation Links

To help identify the exact conditions where a Route Leak Initiation can occur, it is helpful to exclude Link Classifications where it is axiomatically impossible to cause a Route Leak Initiation.

Since a Transit Classification, by definition, can receive all routes, a Transit Link cannot be the source of a Route Leak Initiation. By the same logic, a Special (e.g. Mutual Transit) Classification cannot be the source of a Route Leak Initiation.

This leads to a more precise definition of a Route Leak Initiation.

7. Route Leak Initiation

Route Leak Initiation: A Non-Customer Route which is received over a Peer or Customer Link.

8. Route Leak

Route Leak: any Route where, somewhere in the Path, a Non-Customer Route was received over a Peer or Customer Link. (This is synomous with "was sent over a Peer or Transit Link".)

It should be observed that a route which is not a route leak, has an as-path that matches the following pattern:

{C|S}*P?{T|S}*

Where C is Customer, T is Transit, P is Peer, and S is Special, and "{ | }" denotes either/or, "*" means zero or more occurrences of, and "?" means zero or one occurrences of.

9. Security Considerations

None per se.

10. IANA Considerations

This document contains no IANA-specific material.

11. Acknowledgements

To be added later.

12. References

12.1. Normative References

[RFC1033] Lottor, M., "Domain administrators operations guide", RFC 1033, November 1987.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", RFC 5011, September 2007.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

12.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author’s Address

Brian Dickson
Brian Dickson
703 Palmer Drive,
Herndon, VA 20170
USA

Email: brian.peter.dickson@gmail.com

sidr B. Dickson
Internet-Draft Brian Dickson
Expires: September 7, 2012 March 6, 2012

Route Leaks -- Requirements for Detection and Prevention thereof
draft-dickson-sidr-route-leak-reqts-02

Abstract

The Border Gateway Protocol, version 4, (BGP4) provides the means to advertise reachability for IP prefixes. This reachability information is propagated in a peer-to-peer topology. Sometimes routes are announced to peers for which the local peering policy does not permit. And sometimes routes are propagated indiscriminantly, once they have been accepted.

This document is a requirements document for detection of (and prevention of) route leaks.

Together with the definitions document, it is intended to suggest solutions which meet these criteria, and to facilitate evaluation of proposed solutions.

The fundamental objective is to "solve the route leaks problem".

Author’s Note

Intended Status: Informational.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Rationale

This document analyzes the particulars of situations which introduce route leaks, or propagates those leaks.

Using the definitions previously established, those conditions are reduced to a minimum set of requirements for the identification of route leaks.

Those conditions are validated at length, and all of the assumptions stated, and consequential conditions enumerated.

The result is a set of criteria for solving the route leak problem, preventing any single source of leakage regardless of intent or nature (operator, implementor, bad actor).

1.2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.3. Terminology

The reader is assumed to be familiar with the IETF.

2. Peering Terms and Symbols

We can represent the per-link peering categorizations with the following symbols:

Neighbor is:

a. Transit Provider - T

b. (Transit) Customer - C

c. Peer - P

d. Mutual Transit

In any neighbor relationship, the roles of the parties on either end of the link would be:

T-C

C-T

P-P

Mc-Mtp

Mtp-Mc

(where the last two, Mc/Mtp are a semantic and/or coloring distinction on routes, rather than two separate links.)

3. Local Non-Leak Prefix Advertisement Matrix & Rules

The following matrix shows what prefixes from a given source peering relationship, may be advertised to a given neighbor peering relationship without causing a route leak.

Src \ Dest	P	T	Mtp	Mc	C
P	-	-	-	Y	Y
T	-	-	-	Y	Y
Mtp	-	-	-	Y	Y
Mc	Y	Y	Y	-	Y
C	Y	Y	Y	-	Y

Grouping the like items (by row and column) we get:

Src \ Dest	T/Mtp	P	Mc	C
T/Mtp	-	-	Y	Y
P	-	-	Y	Y
C/Mc	Y	Y	-	Y

When a prefix is sent to any T neighbor, the receiving neighbor sees it as C. Similarly, Mc is seen at Mtp.
The inverse of these is also true: C->T, Mtp->Mc.
And lastly, a prefix sent to a (P) will be received by the neighbor

as a (P).

This means that once a prefix has been sent to any of the two type sets "P" or "C/Mc", it must only subsequently be sent to "C" or "Mc" types.

This results in the regular expression for a valid (non-leaked) path:

Origin - (T - |Mtp -)*(P -)?(C - |Mc -)* Destination

Thus we have the basis for a simple set of rules, which would enable detecting and preventing route leaks.

4. Route Leak Detection Requirements

Based on the advertisement rules, we now have enough information to specify the main rules that a Route Leak Detector would need to observe.

4.1. Coloring Rules

In no particular order, here are the requirements for coloring the path of a route.

- Every BGP peering session (Link) MUST have a type associated with it.
- Neighbors Agree - both sides of a BGP peering link must negotiate and agree on the link type.
- Last Color Agrees with Link - the last color applied to the route must be the consistent with the link type.
- If the Color used towards "Transit" is "Green", and the Color used towards "Peer" or "Customer" is "Yellow", then:
    - The entire Path must have a corresponding set of Colors, one for each AS-Hop.
    - The Path must be of the form (Green)*(Green|Yellow)(Yellow)*.
    - Once a Path has switched to Yellow, it cannot switch back to Green.

- Routes sent to T neighbors must mark the path Green.
- Only Green Routes may be sent to T or P neighbors.
- Routes sent to C or P neighbors must mark the path Yellow.
- A route learned via a P neighbor must be all Green followed by a single Yellow.
- A route learned via a T neighbor must be zero or more Greens followed by one or more Yellows.
- A route learned via a C neighbor must be one or more Greens (and no Yellows).
- Mutual Transit links must preserve the current color.
- Colors may be explicitly marked, or may be inferred as long as there is no room for ambiguity.

4.2. Route Modification Rules

In addressing accidental route leaks, the secondary goal is to also prevent malicious route leaks.

The only additional rule for this is, that any additional BGP attributes implementing this would need to be included in the set of things cryptographically signed. This provides tamper evidence and prevention of substitution of values (on received routes).

This means that the assigning of colors must be handed by implementation based only on Link Type (and current Route color), with no over-ride by the operator possible, with a single exception: It should always be possible to "demote" a route from Green to Yellow, locally before or while sending.

Similarly, route-leak filtering of routes on both the send and receive direction, MUST be done based only on color vs link type. There cannot be an operator-exposed over-ride.

For an operator who has a need to make a routing announcement that violates the Link Type, the correct course of action would be to change the Link type. This would need to be done cooperatively with the party at the other end of the link.

4.3. Single Party Rules

One objective in preventing Route Leaks from being initiated or propogated, is to examine the control points of the routing path itself.

By treating this as a path where the goal is to avoid any single point of failure, we can derive additional rules.

Here, the term "failure" is synonymous with "route leak". In other words, are their any points where a single error or omission can cause a route leak?

If there are any, the goal should be to replace those with equivalent elements which would require two errors or actions, by independent parties, to cause a route leak.

Here are some of the places where this is accomplished or needs to be done by solutions:

- Sender/Receiver - both ends of a link need to agree on the type. Unilateral error here must fail "safe" -> BGP does not establish, with errors.
- Always Validate Color Rules - while the blocking of leaked routes should occur automatically at the point of leak, failure to block a leak SHOULD be detected and the route SHOULD be blocked by the next recipient.

5. Security Considerations

None per se.

6. IANA Considerations

This document contains no IANA-specific material.

7. Acknowledgements

To be added later.

8. References

8.1. Normative References

[RFC1773] Traina, P., "Experience with the BGP-4 protocol", RFC 1773, March 1995.

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, August 1996.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

8.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author’s Address

Brian Dickson
Brian Dickson
703 Palmer Drive,
Herndon, VA 20170
USA

Email: brian.peter.dickson@gmail.com

sidr B. Dickson
Internet-Draft Brian Dickson
Expires: September 7, 2012 March 6, 2012

Route Leaks -- Proposed Solutions
draft-dickson-sidr-route-leak-solns-01

Abstract

The Border Gateway Protocol, version 4, (BGP4) provides the means to advertise reachability for IP prefixes. This reachability information is propagated in a peer-to-peer topology. Sometimes routes are announced to peers for which the local peering policy does not permit. And sometimes routes are propagated indiscriminantly, once they have been accepted.

This document considers the situations that can lead to routes being leaked, and tries to find acceptable definitions for describing these scenarios.

The purpose of these definitions is to facilitate discussion on what a route leak is, and what the scope of the problem space for route leaks is. This, in turn, is intended to inform a requirements document for detection of (and prevention of) route leaks. And finally, the definitions and requirements are intended to allow proposed solutions which meet these criteria, and to facilitate evaluation of proposed solutions.

The fundamental objective is to "solve the route leaks problem".

Author’s Note

Intended Status: Standards track.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Rationale

This document describes two different schemes for implementing a solution for route leaks.

They represent different trade-offs between simplicity of implementation, versus embedding information. The information embedded can be inferred currently from a variety of sources, so the risk/cost of doing so is marginal.

Either solution would be adequate to solve the route leak problem.

Due to the requirement for mandatory establishment of peering link types, and cryptographic protection, the ideal time and place to implement this would be coincident with BGPSEC.

Including route leak protection with BGPSEC may be beneficial to the latter. It is more compelling to deploy a solution to both sets of problems, than to deploy a solution to one or the other alone.

1.2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.3. Terminology

The reader is assumed to be familiar with the IETF.

2. Prefix Attribute Possibilities

If we presume that there are two possible colors for a prefix, then we have three ways to express those colors:

1. A single bit, with two possible values, always attached to a prefix.
2. An attribute whose presence signals one of the colors, and whose absence signals the other color.
3. The same as 2, but with the other color being signaled.

For sake of clarity, we will use a fairly universally understood pair of colors, "green" (meaning "proceed"), and "yellow" (meaning "caution").

So, the three ways of marking the colors are:

Use a green/yellow bit (green if 1, yellow if 0)

Use a "green" attribute (green if present, yellow otherwise)

Use a "yellow" attribute (yellow if present, green otherwise).

Since information is leaked for both the "green/yellow bit" and "yellow attribute", there is no reason to discuss the "yellow attribute" option. It is inferior to both other methods.

3. Encoding Color via Choice of Algorithm

Here, we are presuming that BGPSEC is in use on prefixes, and that BGPSEC includes an explicit algorithm identifier. Currently, the identifier only specifies which algorithm to use to validate the signature in the signature block.

This would be augmented so that for any given algorithm, two identifiers would be assigned. One would be the identifier signifying "Green", and the other would signify "Yellow". When sending a "green" route, the current "green" algorithm would be used. When sending a "yellow" route, the current "yellow" algorithm would be used. Validation would work as usual, with the additional ability to validate the color rules for preventing route leaks.

No additional changes to the structure of the BGPSEC protocol or wire format are needed.

However, there is the leak of information about transit relationships, which is unavoidable with this design.

Routes which violate the path coloring rules but otherwise validate, would be blocked. (They should not occur, but should be checked regardless.) Routes which do not validate under BGPSEC would be blocked regardless, also preventing a potential source of route leaks.

4. Encoding Color via a Second Signature Block

A signature block analogous to the AS-PATH signature block, would be included on any announcement that is "green". The local sender would add her signature to the signature block on these "green" announcements. In addition, the new signature block would be sent across the "green/yellow" boundary to any Peer. However, when sending across the "green/yellow" boundary, would not add her signature to the block.

The recipient would be able to validate all the "green" signatures up to the sender, and if present, the sender’s signature as well. If the "green" signature does not include the sender, no more signatures can be attached. When sending to a "yellow" peer, the "green attribute" block is stripped (if present). The absence of a "green block" means the prefix is considered "yellow". This mechanism is not "free" in that more crypto calculations are needed, the structure of the BGPSEC attributes change, and more data is needed on each announcement within the "green" zone.

However, no information concerning relationships is leaked, beyond what the recipient can already infer. A transit provider already knows his/her customers, and their customers, etc.

From a scaling perspective, it should be noted that only customers’ prefixes require additional signatures, so the number of prefixes with those signatures is proportionally smaller. Signature validation is only done on the "green block" upon receiving a customer’s routes or a peer’s routes. This also minimizes the incremental cost.

Since it is physically impossible to promote a "yellow" route to a "green" route, because the originators "green" block is absent, this is a very strong mechanism for stopping route leaks. Validating link type versus color, after validation of any "green block" present, is sufficient to stop route leaks.

5. Security Considerations

None per se.

6. IANA Considerations

This document contains no IANA-specific material.

7. Acknowledgements

To be added later.

8. References

8.1. Normative References

[RFC1773] Traina, P., "Experience with the BGP-4 protocol", RFC 1773, March 1995.

[RFC1997] Chandrasekeran, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, August 1996.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

8.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author’s Address

Brian Dickson
Brian Dickson
703 Palmer Drive,
Herndon, VA 20170
USA

Email: brian.peter.dickson@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 1, 2012

J. Gersch
Secure64 SW Corp
D. Massey
Colorado State University
E. Osterweil
Verisign

February 29, 2012

Reverse DNS Naming Convention for CIDR Address Blocks
draft-gersch-dnsop-revdns-cidr-01.txt

Abstract

The current reverse DNS naming method is used to specify a complete IP address. There currently is no standard way for it to handle address ranges; for example, there is no formal mechanism for specifying a reverse DNS name for the block of addresses specified by the IPv4 prefix 129.82.0.0/16. Defining such a reverse DNS naming convention would be useful for a number of applications. These include applications for secure BGP routing, and applications that need host-information for a device owning a complete IPv6 address block. This draft proposes a naming convention for encoding CIDR address blocks in the reverse DNS.

Status of this Memo

This Internet-Draft will expire on September 1, 2012.

Copyright Notice

Table of Contents

1. Introduction

This draft proposes a common naming convention for entering CIDR prefixes into the Reverse DNS.

The Reverse DNS provides a naming convention for both IPv4 and IPv6 addresses. At this time, the most common use of the reverse-DNS is to associate an IP address with a PTR resource record that identifies the corresponding host name. For example, IP address 129.82.138.2 is encoded as 2.138.82.129.in-addr.arpa and a PTR resource record identifies the host name as alpha.netsec.colostate.edu. The Reverse DNS would be more expressive if we had a formal convention for encoding and returning information associated with a network address range, not just a unique IP address. For example, one would like to store and resolve resource records associated with a prefix range such as 129.82.128/17.

Given such a capability, a variety of new applications and services would be enabled. For example, internet routing operators could publish authorized BGP route origins for their network address blocks in the reverse-DNS as proposed in [I-D.gersch-grow-revdns-bgp]. Another application could query for a set of host-names or services associated with an address block; for example, to indicate the authorized mail servers for an address block.

Yet another interesting possibility is to solve a problem with IPv6 dynamic DNS assignments. In IPv4, the owner of address block could simply include one PTR record for every available address. In fact, ISPs commonly pre-populate the reverse DNS zone for their customers. However, this approach clearly does not scale for IPv6 where the number of addresses becomes excessively large. For example, allocation of a /48 (not uncommon in IPv6) includes 2^80 addresses and notes adding 1000 PTR records per second would require over 38 trillion years to pre-populate the reverse DNS [I-D.howard-isp-ip6rdns]. The ability to name prefix blocks rather than individual addresses could help address this problem by publishing records associated with an entire IPv6 address range instead of replicating or synthesizing answers to unique address queries.

The above list of possible applications is not intended to be complete, but instead suggest some of the possibilities.

1.1. Aligning the DNS and IP Hierarchies

A key observation is that both the DNS names and IP addresses are part of a hierarchical tree structure and any naming convention should respect and align these tree structures.

In the DNS hierarchical tree structure 128.82.129.in-addr.apra is logically below 82.129.in-addr.apra, which is logically below 129.in-addr.arpa. Other "flat" approaches to naming, such as Distributed Hash Tables, have been proposed, but the DNS tree structure remains a powerful abstraction. It forms the basis for the operation of DNS; caching, delegation, DNSSEC signing, and so forth all benefit from the DNS tree structure.

IP addresses also have a logical tree structure where 129.82.128.0/24 is subprefix (logically below) 129.82.0.0/16 which is a subprefix of 129.0.0.0/8. The reverse DNS aligns with the structure; 128.82.129.in-addr.arpa is logically below 82.129.in-addr.arpa which is logically below 129.in-addr.arpa. This alignment between the DNS hierarchy and the IP address hierarchy serves both systems well and allows one to easily encode prefixes that fall on an octet boundary (e.g. IPv4 prefixes whose mask length is a multiple of 8).

The challenge is to preserve this alignment even when even when CIDR prefixes do not fall on octet boundaries. For example, 129.82.128.0/19 is a subprefix of 129.82.128.0/18. The DNS name for 129.82.128.0/19 should be logically below the DNS name for 129.82.128.0/18. This document introduces a naming convention for CIDR prefixes that restores this alignment.

1.2. Purpose

In order to enable these applications, one must map an IPv4 or IPv6 prefix into a reverse-DNS name. There are various subtleties, advantages and disadvantages that emerge when trying to define a naming convention. Today, zone administrators can use their own individual approaches to encode a prefix in the reverse DNS. This requires no DNS protocol changes and no modifications to resolvers, caches, or authoritative servers. The emergence of different encoding standards complicates (but does not prevent) the design of systems that would make use of these resource records. The aim of this work is to introduce a standard convention.

1.3. Terminology

The following terms are used throughout out the document:

Reverse DNS:
We use the term Reverse DNS to refer to the domains in-addr.arpa and ip6.arpa.

Prefix:

A prefix refers to IPv4 or IPv6 address range specified by a network portion and mask length, as described in [RFC4632]. For example, 129.82.0.0/16 and 129.82.128/18 are examples of IPv4 prefixes.

Octet Boundary:

An IPv4 prefix falls on an octet boundary if its mask length is a multiple 8. For example, 129.82.0.0/16 is on an octet boundary while 129.82.128/18 does not fall on octet boundary. Prefixes that are on octet boundary naturally map to the reverse DNS. Prefixes that are not on octet boundary are more complex and the main challenge for any naming convention.

2. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Design Requirements

A naming convention to specify CIDR address blocks in the reverse-DNS has several design goals:

1. Autonomy: The owner of a reverse-DNS zone file associated with a CIDR address block must be able to act independently from any other organization in order to create or modify data records within the DNS zone.

2. Coverage Authority: With the exception of data that has been sub-delegated to a child zone, the reverse DNS zone must be authoritative for all sub-prefixes below the covering prefix. Any query for a sub-prefix must be answered with a data record or NXDOMAIN specifying this zone as the authority.

3. Allow Delegation: It must allow the zone owner to delegate smaller address blocks to a child zone which will be independently managed.

4. Conformance: It should align with naming conventions and delegation structures already in use by the RIR’s for IN-ADDR.ARPA and IP6.ARPA.

5. Simplicity: The naming structure should be understandable, or at a minimum, able to be easily constructed by software provisioning tools and utilities such as DIG.

4. Related Work

The process of mapping CIDR addresses into the reverse-DNS name space is difficult because the prefix length of an IPv4 CIDR address is an arbitrary number from 0 to 32. These numbers do not necessarily align with an IPv4 octet.

4.1. CIDR Naming via RFC 2317

Since CIDR address no longer align with octet boundaries, the CIDR specification in [RFC4632] notes that there is "some increase in work for those who maintain parts of the IN-ADDR.ARPA zone." [RFC2317] is offered as a technique to populate IN-ADDR.ARPA. The intent of this work is to encode IPv4 addresses and the approach is designed to "address spaces covering fewer than 256 addresses."

Suppose organization A owns 129.82.138.0/30. This address space covers four IPv4 addresses; namely 129.82.138.0, 129.82.138.1, 129.82.138.2 and 129.82.138.3. Giving organization A control of the reverse zone "138.82.129.in-addr.arpa." would allow Organization A to enter PTR resource records for each of its 4 addresses. However, it also gives organization A the ability to enter PTR resource records for 252 other IP addresses from 129.82.138.4 to 129.82.138.255. These addresses are managed by other organizations. Sharing the 138.82.129.in-addr.arpa between multiple organization is not practical and creating a seperate zone for each IP address (e.g. creating the zone 0.138.82.129.in-addr.arpa) is very high overhead to store a single PTR record.

[RFC2317] addresses this problem by creating CNAME records in 138.82.129.in-addr.arpa zone. Organization A administers a zone named 0/32.138.129.in-addr.arpa. CNAME records in the 138.82.129.in-addr.arpa zone point to entries in Organization A’s 0/32.138.82.129.in-addr.arpa zone. For example, 1.138.82.129.in-addr.arpa. is a CNAME pointing to 1.0/32.138.82.129.in-addr.arpa. A full description is found in [RFC2317].

This approach was not intended to encode IP address for address spaces smaller than a "/24". It was not intended for encoding prefixes. It does not specify how one might encode a prefix and it is not trivial to extend this approach to CIDR prefixes. In particular, the design requirements of Coverage Authority, Allowing Delegation, and arguably Simplicity are not easily met by extending the RFC to included prefixes.

4.2. Prior Work on CIDR Names for Routing

Over a decade ago, [I-D.bates-bgp4-nlri-orig-verif] proposed to use the reverse DNS to verify the origin AS associated with a prefix. This requires both a naming convention for converting the name into a prefix and additional resource record types for storing origin information, along with recommendations on their use.

Our focus in this draft is on the naming convention. Draft [I-D.bates-bgp4-nlri-orig-verif] as well as other subsequent work on BGP security, extends [RFC2317] style names to encode a prefix. For example, the draft proposes to encode the prefix 10.1.128/20 as the DNS name 128/20.1.10.bgp.in-addr.arpa.

In [I-D.bates-bgp4-nlri-orig-verif], the DNS hierarchy and the IP address hierarchy diverge and the approach fails to meet the Coverage Authority requirement. To see this, consider the prefixes 10.1.128/20 and 10.1.128/21. in CIDR terminology, 10.1.128/21 is covered by 10.1.128/20, but this relationship is not captured in the DNS hierarchy. 10.1.128/21 is encoded as 128/21.1.10.bgp.in-addr.arpa and thus 10.1.128/20 and 10.1.128/21 are siblings in the DNS tree structure.

This can be overcome by introducing a large number of CNAME records; one for every potential subprefix. We instead provide an approach where the CIDR hierarchy and DNS hierarchy align.

5. Reverse DNS CIDR Name Specification

The naming method described in this section is based on the well-known technique of ANDing a bit-mask with the low-order octet of an IP address. The binary result is then broken up into individual sub-names using the "." separator. The result looks like an ENUM or IPv6 reverse-DNS address; that is, a string of chained empty non-terminal sub-names.

This name-chaining creates the desired effect of being able to allow a DNS zone delegation at any point in the chain. The naming scheme allows the creation of two /17’s from a /16, two /18’s from a /17, and so on.

5.1. IPv4 Address Block Naming

The CIDR to Reverse-DNS naming convention works as follows:

1. Remove any octets that are not significant. An octet is signficant if it includes any part of the network address. An octet is not significant if all bits correspond to the host portion of the address. For example, 129.82.0.0/16 --> 129.82 and 129.82.160.0/19 --> 129.82.160

2. Calculate N where N = prefix_length mod 8. If N equals 0, invert the address and add in-addr.arpa, per the usual reverse-DNS method; 129.82 --> 82.129.in-addr.arpa.

3. If N is not equal 0, the prefix is not on an octet boundary and we perform the following name construction:

    A. Truncate the name to remove the least significant octet. Add a "m" label to this domain name to indicate "mask".

    B. Convert the least significant octet to binary, separating each bit into its own label (with a "." character).

    C. Truncate the binary labels to the N significant labels that correspond to the given prefix_length.

    D. Reverse the string and add ".in-addr.arpa."

Several examples illustrate this algorithm. These examples show the conversion to binary, followed by the truncation, followed by the name reversal.

```
129.82.0.0/16   --> 82.129.in-addr.arpa. (at octet boundary)
```

```
       129.82.64.0/18      --> 129.82.m.0.1.0.0.0.0.0.0
                       --> 129.82.m.0.1 (N = 18 mod 8 = 2)
                       --> 1.0.m.82.129.in-addr.arpa.


       129.82.64.0/20  --> 129.82.m.0.1.0.0.0.0.0.0
                           --> 129.82.m.0.1.0.0  (N = 20 mod 8 = 4)
                       --> 0.0.1.0.m.82.129.in-addr.arpa.


           129.82.160.0/20 --> 129.82.m.1.0.1.0.0.0.0.0
                           --> 129.82.m.1.0.1.0 (N = 20 mod 8 = 4)
                           --> 0.1.0.1.m.82.129.in-addr.arpa.


           129.82.160.0/23 --> 129.82.m.1.0.1.0.0.0.0.0
                           --> 129.82.m.1.0.1.0.0.0.0 (N = 23 mod 8 = 7)
                           --> 0.0.0.0.1.0.1.m.82.129.in-addr.arpa.


           15.192.0.0/12   --> 15.192.m.1.1.0.0.0.0.0.0
                           --> 15.192.m.1.1.0.0    (N = 12 mod 8 = 4)
                           --> 0.0.1.1.m.15.in-addr.arpa.
```

The conversion from a reverse-DNS name back to CIDR is simple. First calculate the prefix length from the name using the formula:

```
        plen = 8*(count of full octets) + (count of binary digits)
```

Then reverse the string, add up the values of the binary digits to build a final octet, then append a "/" and the prefix length.

Examples:

```
        1.0.m.82.129.in-addr.arpa  --> 129.82.64.0/18
        (example has 2 octets + 2 binary digits, so mask length = 18)


    0.0.1.0.m.82.129.in-addr.arpa --> 129.82.64.0/20
        (example has 2 octets + 4 binary digits, so mask length = 20)


        0.0.0.1.0.1.m.129.in-addr.arpa--> 129.160.0/14
        (example has 1 octet + 6 binary digits, so mask length = 14)
```

5.2. IPv6 Address Block Naming

The IPv6 naming convention is similar, with the exception that 4-bit nibble boundaries are used instead of octets, the mod calculation is based on 4 instead of 8, and "ip6.arpa" is used as the suffix.

Examples:

```
        2607:fa88::/32     --> 8.8.a.f.7.0.6.2.ip6.arpa
           (on nibble boundary)


        2607:fa88:8000::/33 --> 2.6.0.7.f.a.8.8.m.1.0.0.0
                           --> 2.6.0.7.d.a.8.8.m.1    (33 mod 4 = 1)
                           --> 1.m.8.8.a.f.7.0.6.2.ip6.arpa


        2607:fa88:e000::/35 --> 2.6.0.7.f.a.8.8.m.1.1.1.0
                           --> 2.6.0.7.d.a.8.8.m.1.1.1(35 mod 4 = 3)
                           --> 1.1.1.m.8.8.a.f.7.0.6.2.ip6.arpa
```

5.3. An Alternative Encoding For Names at Octet Boundaries

If a prefix is on an octet boundary, the algorithm stops at step 2. However by applying Step 3 of the algorithm, one could also obtain an alternate encoding for the same prefix For example, applying the algorithm produces the standard encoding 129.82.1.0/24 --> 1.82.129.in-addr.arpa. If one applies step 3 of the algorithm, one gets the alternate encoding 129.82.1.0/24 --> 1.0.0.0.0.0.0.0.m.82.129.in-addr.arpa.

Deployment experience has shown that the alternate encoding can be very useful in some circumstances. To see this, consider the case where an organization owns the prefix 129.82.0.0/16. The organization’s central IT office administers the 82.129.in-addr.arpa zone. The central IT office has delegated 1.82.129.in-addr.arpa to a remote division. The remote division is capable of managing PTR records within this zone, but lacks the technical expertise to manage records associated with the prefix 129.82.1.0/24. For example, the remote division should not be authorizing mail servers, announcing BGP routes, or other prefix related tasks.

Unfortunately for the central IT office, it is sometimes useful to store information at the 129.82.1.0/24 prefix. For example, the central IT office may want to add a record listing the authorized mail servers for this prefix or indicate the prefix cannot announce BGP routes. The problem is that these records would be stored at the name 1.82.129.in-addr.arpa, which is managed by the remote subdivision. Rather than ask the remote division to enter these resource records, the central IT office would like to handle this taks for the remote division.

By exploiting the alternate encoding, the central IT office can store and manage records at the name 1.0.0.0.0.0.0.0.m.82.129.in-addr.arpa. The key distinction is that this alternate encoding of the name is part of the 82.129.in-addr.arpa zone. This allows the central IT organization to administer resource records on behalf of the remote division and greatly simplifies some operations.

The following rules apply to the alternate encoding:

An application MUST first try the standard encoding of the name.

If the requested resource record type is not found at the standard encoding of the name and the prefix is at an octet boundary, an application SHOULD try the alternate encoding.

If the same resource record type is present at both the standard encoding and the alternate encoding, the RRSet at the standard encoding of the name MUST take precedence.

Finally, note that the alternate encoding allows a parent zone to create RRSets on behalf of a child zone. If an RRSet exists at both the parent and the child, the child’s RRset takes precedence. In other words, the parent can enter data on behalf of a child but the child can always over-ride the parent.

Examples:

```
129.82.160.0/24 --> 129.82.m.1.0.1.0.0.0.0.0
        --> 0.0.0.0.0.1.0.1.m.82.129.in-addr.arpa.


129.82.255.0/24 --> 129.82.m.1.1.1.1.1.1.1.1
        --> 1.1.1.1.1.1.1.1.m.82.129.in-addr.arpa.


2607:fa88:e000::/36 --> 2.6.0.7.f.a.8.8.m.1.1.1.0
                   --> 0.1.1.1.m.8.8.a.f.7.0.6.2.ip6.arpa
```

6. Security Considerations

This document only introduces a naming convention. Applications that make use of this naming convention may require the use of DNSSEC to validate the resource records stored at these names.

7. IANA Considerations

This document does not request any IANA action.

8. Acknowledgments

The authors would like to thank Danny McPherson (Verisign), Lixia Zhang (UCLA), and Kim Claffy (CAIDA) for their comments and suggestions. This document was aided via numerous discussions at NANOG, IETF and private meetings with ISPs, telecomm carriers, and research organizations too numerous to mention by name. Thanks to all for your comments and advice.

9. Change History

Changes from version 00 to 01

Introduction added an additional subsection on aligning the DNS hierarchy with the IP address hierarchy.

Clarified step 1 of the naming algorithm on removing octets that are not signficant.

Expanded and clarified the discusion of alternate name encodings for prefixes on an octet boundary.

Added Eric Osterweil as a co-author

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

10.2. Informative References

[I-D.bates-bgp4-nlri-orig-verif]
Bates, T., Bush, R., Li, T., and Y. Rekhter, "DNS-based NLRI origin AS verification in BGP", draft-bates-bgp4-nlri-orig-verif-00 (work in progress), January 1998.

[I-D.gersch-grow-revdns-bgp]
Gersch, J., Massey, D., Osterweil, E., and L. Zhang, "DNS Resource Records for BGP Routing Data", draft-gersch-grow-revdns-bgp-00 (work in progress), February 2012.

[I-D.howard-isp-ip6rdns]
Howard, L. and A. Durand, "Reverse DNS in IPv6 for Internet Service Providers", draft-howard-isp-ip6rdns-04 (work in progress), September 2010.

[RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 2317, March 1998.

Appendix A. Example Zone Files

A.1. Example 1

This example shows several DNS records added to an existing reverse-DNS zone file at octet boundary 129.82.0.0/16. The records show how BGP route origins for a CIDR prefix could be specified in the zone file. Otherwise no other changes were made. This example has added records with routing information pertinent to address blocks 129.82/16 and the four /18’s at 129.82.0.0/18, 129.82.64.0/18, 129.82.128.0/18, and 129.82.192.0/18.

Note: this internet draft is not proposing the RRTypes for routing shown here; they are only presented as sample content for the proposed naming convention. A separate document [I-D.gersch-grow-revdns-bgp] provides details on these RRTYpes.

In addition, the example shows a record for a /24 using the full 8-bit alternate encoding (Section 5.3) so that the data can be placed in this parent zone rather than in the child zone at 177.82.129.in-addr.arpa.

```
$TTL 3600
$ORIGIN 82.129.in-addr.arpa.

@    IN    SOA     rush.colostate.edu.  dnsadmin.colostate.edu. (
                        2012021300      ; serial number
                        900             ; refresh, 15 minutes
                        600             ; update retry, 10 minutes
                        86400           ; expiry, 1 day
                        3600            ; minimum, 1 hour
                       )

     IN    NS      dns1.colostate.edu.
     IN    NS      dns2.colostate.edu.

@                  IN   TYPE65400 \# 0
;                  RLOCK    deny all route announcements
;                           except those authorized

@                  IN   TYPE65401 \# 4 00002f71
; 129.82.0.0/16         SRO 12145   (SRO "Secure Route Origin")

0.0.m              IN   TYPE65401 \# 4 00002f71
; 129.82.0.0/18         SRO 12145

1.0.m              IN   TYPE65401 \# 4 00002f71
; 129.82.64.0/18        SRO 12145

0.1.m              IN   TYPE65401 \# 4 00002f71
; 129.82.128.0/18       SRO 12145

1.1.m              IN   TYPE65401 \# 4 00002f71
; 129.82.192.0/18       SRO 12145

1.0.0.0.1.1.0.1.m  IN   TYPE65401 \# 4 00004070
; 129.82.177.0/24       SRO 12145

;  delegations required for 256 /24 zones which contain PTR records

1   IN  NS  dns1.colostate.edu.
    IN  NS  dns2.colostate.edu.
2   IN  NS  dns1.colostate.edu.
    IN  NS  dns2.colostate.edu.

;  continuation to 255 is left out for the sake of brevity
```

A.2. Example 2

This example illustrates the creation of a new zone for 216.17.128.0/17 which is not at an octet boundary. The existing 256 zones delegated at IN-ADDR.ARPA for the range 0.17.128 through 255.17.216.in-addr.arpa remain unchanged; they contain PTR records maintained by the appropriate zone owners.

In this example we have added several records all at the same domain name with information pertinent to address block 216.17.128.0/17.

Only a single new delegation needs to be added to IN-ADDR.ARPA:

```
        1.m.17.216.in-addr.arpa  NS   ns.frii.net
```

This delegation refers to the new /17 zone and is not in conflict with any of the pre-existing /24 zones.

```
   $TTL 3600
   $ORIGIN 1.m.17.216.in-addr.arpa.

   @    IN   SOA     ns1.frii.net.  hostmaster.frii.net. (
                           2012021300      ; serial number
                           14400           ; refresh, 4 hours
                           3600            ; update retry, 1 hour
                           604800          ; expiry, 7 days
                           600             ; minimum, 10 minutes
                          )

        IN   NS      ns1.frii.net.
        IN   NS      ns2.frii.net.

   $ORIGIN 17.216.in-addr.arpa.

   1.m               IN   TYPE65400 \# 0
   ;                 RLOCK    deny all route announcements
   ;                          except those authorized

   1.m               IN   TYPE65401 \# 8 000019b6
   ; 216.17.128.0/17      SRO 6582   (SRO "Secure Route Origin")

   1.m               IN   TYPE65401 \# 8 000019b6
   ; 216.17.128.0/17      SRO 6582

   ; no other delegations or PTR records are needed in this zone file
```

Authors’ Addresses

Joe Gersch
Secure64 SW Corp
Fort Collins, CO
US

Email: joe.gersch@secure64.com

Dan Massey
Colorado State University
Fort Collins, CO
US

Email: massey@cs.colostate.edu

Eric Osterweil
Verisign
Reston, VA
US

Email: eosterweil@verisign.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 1, 2012

J. Gersch
Secure64 SW Corp
D. Massey
Colorado State University
E. Osterweil
Verisign
L. Zhang
UCLA

February 29, 2012

DNS Resource Records for BGP Routing Data
draft-gersch-grow-revdns-bgp-00

Abstract

This draft proposes the creation of two DNS record types for storing BGP routing information in the reverse DNS. The RLOCK record allows prefix owners to indicate whether the DNS is being used to publish routing data. The SRO record allows operators to indicate whether an IPv4 or IPv6 prefix ought to appear in global routing tables and identifies authorized origin Autonomous System Number(s) for that prefix. The published data can be used in a variety of contexts and can be extended to include additional information. This work is part of an on-going effort and is accessible in an active testbed.

Status of this Memo

This Internet-Draft will expire on September 1, 2012.

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

This draft describes a method in which a prefix owner can exploit the existing reverse DNS tree structure, along with the authentication provided by DNSSEC [RFC4033], to publish information about whether a prefix can be announced and to identify the origin Autonomous System(s) that may originate a route to that prefix. This data is complementary to a variety of other data sources ranging from existing databases to new directions.

Publishing route information in the Reverse DNS takes advantage of infrastructure that already exists and has been globally deployed. No new infrastructure deployment is required, in contrast with approaches that use purpose-built resource certification.

Other key advantages to using the Reverse DNS are that it 1) has been in successful operation for many years, 2) has an existing operational model where prefix owners currently manage their IP address space (through various models from local operation to hosting companies), 3) has an existing operational model where both registries and providers delegate authority to entities receiving address space, 4) the resulting reverse DNS data can be authenticated using DNSSEC [RFC4033], and 5) the data can be easily checked using simple tools ranging from DNS query tools such as DIG to more elaborate systems.

A prefix owner must OPT-IN to the approach. Prefix owners who do not take any action are not impacted, but also do not gain any advantages. Prefix owners that do choose to participate would thereby enable a number of tools to make use of the published data. The objective of this draft is to standardize the format for indicating participation and publishing data. A variety of potential uses for the data are discussed later in the document, but are provided only to illustrate the usefulness of the data and should not be taken as a comprehensive list of all possible applications.

Examples taken directly from the current testbed are included in the appendix.

1.2. Scope

The scope of this internet draft is purposely limited to the subject of BGP route origins. There are many other possible topics that could be explored: BGP path verification, BGP capacity constraints, man-in-the-middle attacks, routing policy, address ownership assignment and provenance, route ingress and egress filtering, interface to internet routing registries, and so on. These are all reasonable extensions.

We limit the scope of this internet draft to the prevention of origin and sub-prefix hijacks -- a capability that can be implemented and deployed in a reasonable time frame. Future expansion is readily made possible: the SRO record is kept simple for now, but may be expanded to incorporate additional fields. New RR types can also be added later for additional capabilities.

The proposed naming structure and record types recommend that a unique entry be published for each prefix, not ranges as with RPKI. This can make routing security policy explicit and help minimize route table bloat.

2. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Overview of Route Publishing

This document defines two new DNS resource records types (RRTypes)

1. The RLOCK RRType (Route Lock)
   - Purpose: Indicates that the Reverse DNS zone has enabled BGP route publishing.
   - The presence of the RLOCK Record at the apex of a Reverse DNS zone indicates that a prefix owner has OPTED-IN to BGP Route Publishing. All route announcements that map to this zone will be denied as BOGUS unless an SRO record exists that specifically authorizes the announcement.
2. The SRO RRType (Secure Route Origin)
   - Purpose: Declare an authorized route origin ASN for comparison against BGP route announcements.
   - Placed in the Reverse DNS at the domain name corresponding to the associated CIDR address block.

Organizations that have been assigned and/or allocated CIDR address blocks also have Reverse-DNS delegations assigned to them from either the Regional Internet Registries (RIPE, ARIN, APNIC, etc.) or from a sub-delegation.

Address-block owners may use these new record types to declare authoritative data for route origins associated with that address block. This data may be declared statically, with a long TTL (Time To Live) if the routing data changes infrequently. Alternatively, dynamic DNS and short TTLs can be used to rapidly publish and disseminate the authoritative information on a world-wide basis in near real-time.

The RLOCK and SRO records are to be stored in the reverse-DNS in zones with domain names that correspond to the associated CIDR address block. These domain names are to be constructed per the naming specification described in [I-D.gersch-dnsop-revDNS-CIDR].

The RLOCK and SRO records MUST be signed with DNSSEC and have a valid DNSSEC chain-of-trust.

4. Overview of Route Verification

Various applications could be written to use BGP records published in the Reverse DNS. One example is an application to perform near-real-time route origin verification that alerts operators of hijacks or directly interacts with a router to prevent the hijack. Another application could perform a nightly analysis that generates router prefix filters. A third application could cross-check data in the Internet Routing Registries (IRR) against the data in the reverse DNS. This list is not intended to be comprehensive, but instead aims to illustrate the potential uses of the published data.

These applications analyze BGP announcements by performing DNS queries to classify route route announcements into one of the following three categories:

1. "VALID": a DNSSEC-validated SRO RRSET was received and one of the route origins in the RRSET matches the origin contained in the BGP route announcement.

2. "BOGUS": a route hijack was detected.

    A. The DNSSEC-validated SRO responses received did NOT match the origin of the route announcement. This is indicative of an origin hijack.

    B. There was no SRO record at the domain name corresponding to this address block, but the authoritative zone did contain an RLOCK statement. This is indicative of a sub-prefix hijacks.

3. "VIABLE": there was no SRO record for this prefix and no RLOCK record to protect the zone, or the data did not properly validate with DNSSEC. In this case, the algorithm cannot authoritatively state that the prefix is valid or bogus, so it is simply marked as viable. Most routes today are in this category, as it takes a specific action to OPT-IN to this methodology.

This verification algorithm MUST "fail-safe". If a query for a DNS record fails, or if DNSSEC fails to validate the record, the algorithm MUST behave as if no DNS records were present in the first place. This results in marking a BGP announcement as "VIABLE". One could completely unplug a router verification application at any time and internet routing would continue to work just as it does today. The default state is always "viable".

Note that this implies the verification algorithm MUST use DNSSEC-enabled queries (set the DO bit) and MUST check for a validated response (the AD bit). A successful DNSSEC-downgrade attack would result in classifying records as "viable". However the redundancy in DNS would allow checking of multiple slave DNS servers should DNSSEC fail to validate.

The core of the verification algorithm can be summarized as follows:

1. Upon receipt of a BGP announcement, perform a DNSSEC-validated query for the SRO records at the domain name corresponding to the CIDR prefix in the BGP announcement.

2. Case 1: If no records exist (NXDOMAIN or NOERROR with number of answers=0), use the AUTHORITY section of the answer to determine the covering zone. Perform a query to that domain name (the zone apex) for an RLOCK record. There are two possible responses to the RLOCK query:

   A. NOERROR, answer=0: the RLOCK does not exist; the zone owner has not opted in. Mark the announcement as "VIABLE".

   B. RLOCK exists: the zone owner has OPTED-IN. Mark the announcement as "BOGUS" since no SRO record exists to vouch for the announcement. This may be an example of a sub-prefix hijack.

3. Case 2: One or more SRO records were returned from the query. Loop through each SRO in the RRSET to compare the origin with the data in the route announcement. If a record with a matching set of data is found, mark the announcement as "VALID". If no match is found, mark the announcement as "BOGUS".

This algorithm can be extended to handle the case of "overlapping" domain names at octet boundaries. Consider the example where a /16 zone has 256 zone delegations for each of its /24 children. For ease of implementation the zone author may wish to place an SRO or RLOCK statement at the overlapping domain name contained in the parent zone rather than create data within the 256 child zones.

In this example, the algorithm should check for BGP data in the /24 zone as normal. If data is found, it is considered authoritative and the algorithm stops. If no SRO or RLOCK is found in this /24 zone, the algorithm queries the "overlapping name" as defined in [I-D.gersch-dnsop-revDNS-CIDR] for an SRO record. If no records are found, it then queries the parent zone (as defined by the AUTHORITY portion of the DNS answer) for an RLOCK statement.

5. The RLOCK Resource Record

The RLOCK resource record indicates "Route Lock". This record is placed at the apex of a reverse-DNS zone to indicate that the zone is being used to publish routing information. If this record is present, all route announcements for the CIDR address block covered by this zone MUST be marked as "bogus" unless they are specifically authorized by a SRO record.

The main purpose of the RLOCK statement is to indicate participation (OPT-IN) and as a side-effect prevent sub-prefix route hijacks. Applications that query for an SRO record may get an NXDOMAIN or NOERROR with 0 answers. In this case, the application queries the domain name specified in the AUTHORITY section for an RLOCK record (this will be at the zone apex). If the RLOCK is present, the route announcement MUST be marked as "bogus". Otherwise there is no SRO and no RLOCK, so the route announcement MUST be marked as "viable" (with the possible exception outlined next regarding "overlapping" octet boundaries).

The RLOCK statement may also be present at zone cuts created at octet or nibble boundaries. The "overlapping domain name" specified in [I-D.gersch-dnsop-revDNS-CIDR] is used to specify the CIDR address block. This type of RLOCK allows the zone author to create one parent zone with 256 delegations to the next octet and add an RLOCK for each one of the child zones. The alternative is to edit all 256 child zones to place the RLOCK at each zone apex. Applications that search for an RLOCK should also search the parent zone to see if there is an RLOCK at the overlapping name.

The effective span of control for an RLOCK is dependent on the structure of the Reverse DNS zone. To be more specific, a Reverse DNS zone that has no delegations will have a span of control that covers all prefixes at or below the CIDR prefix specified by the domain name at the zone apex. Any zone delegation (also known as a "cut point") starts a new zone authority. Those prefixes in the delegated zone will not be covered by the parent zone’s RLOCK. As an example, consider the zone at 129.82.0.0/16 and assume that it has only one delegation at 129.82.138.0/24. The /16 RLOCK covers all prefixes within the /16 to /32 range with the exception of prefixes within the 129.82.138.0/24 through /32 range. The child zone would need to have its own RLOCK, either directly, or with an "overlapping" domain name.

The RLOCK record MUST be signed with DNSSEC and have an associated RRSIG record. If a resolving DNS server cannot validate the DNSSEC signature, the SRO record should be ignored as if it were not even present in the zone.

The Type value for the RLOCK RR type is currently unassigned. We are temporarily using private RRTYPE TYPE65400 until a formal number is assigned by IANA.

The RLOCK RR is class independent.

The RLOCK RR has no special TTL requirements.

Example use of RLOCK records, taken directly from the current testbed, are included in the appendix.

5.1. RLOCK RDATA Wire Format

The RLOCK record contains no RRData (RDLength field = 0).

5.2. RLOCK Presentation Format

Since there is no RRDATA, the presentation format of the RDATA portion is simply the RLOCK keyword with no extra fields.

5.3. RLOCK RR Examples

The following example shows an RLOCK RR enabling routing security for the zone covering 129.82.0.0/16.

```
82.129.in-addr.arpa.  86400  IN    RLOCK
```

The following example shows RLOCK at "overlapping /24" address blocks. The domain name uses the reverse-DNS naming convention for CIDR address blocks specified in [I-D.gersch-dnsop-revDNS-CIDR].

```
0.0.0.0.0.0.0.0.m.82.129.in-addr.arpa.  86400  IN    RLOCK
0.82.129.in-addr.arpa.  86400  IN  NS   ns1.org.edu
1.0.0.0.0.0.0.0.m.82.129.in-addr.arpa.  86400  IN    RLOCK
1.82.129.in-addr.arpa.  86400  IN  NS   ns1.org.edu
0.1.0.0.0.0.0.0.m.82.129.in-addr.arpa.  86400  IN    RLOCK
2.82.129.in-addr.arpa.  86400  IN  NS   ns1.org.edu
1.1.0.0.0.0.0.0.m.82.129.in-addr.arpa.  86400  IN    RLOCK
3.82.129.in-addr.arpa.  86400  IN  NS   ns1.org.edu

     . . . Continuing to

1.1.1.1.1.1.1.1.m.82.129.in-addr.arpa.  86400  IN    RLOCK
255.82.129.in-addr.arpa.  86400  IN  NS   ns1.org.edu
```

6. The SRO Resource Record

Zones that participate in this approach use "Secure Route Origin" (SRO) resource records to indicate that a prefix may be announced. This record contains a mandatory ORIGIN ASN field. Both 32 and 64 bit AS numbers are accommodated.

The ORIGIN AS indicates an AS number that is authorized to originate a route announcement for the CIDR address block associated with the SRO record’s Reverse DNS domain name.

The SRO record MUST be signed with DNSSEC [RFC4033] and have an associated RRSIG record. If a resolving DNS server cannot validate the DNSSEC signature, the SRO record should be ignored and an attempt should be made to query an alternate DNS server. If all servers fail, the route prefix should be classified as "VIABLE".

The Type value for the SRO RR type is currently unassigned. We are temporarily using TYPE65401 until a formal number is assigned by IANA.

The SRO RR is class independent.

The SRO RR has no special TTL requirements.

6.1. SRO RDATA Wire Format

The SRO RDATA wire format MUST contain a minimum of 4 octets which specify the ORIGIN AS number. 2-octet AS Numbers MUST be encoded with leading zeroes to construct a complete 4-octet field.

The SRO record type is intended to evolve over time; in the future there may be optional extensions to indicate a version numbers and other fields such as last hop, system capacity, IRR information, etc. The value of the RDLength provides the flexibility to determine whether additional fields are present or not. In this first version of the SRO record, the the RDLENGTH will be 4. Applications MUST always interpret the first 4 octets as the ORIGIN AS number.

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         ORIGIN AS Number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.2. SRO RRDATA Presentation Format

The presentation format of the RDATA portion is as follows:

AS Numbers are represented in asdot notation which is a combination of asplain and asdot+ notation. That is, any ASN in the 2-octet range is represented in asplain (simple decimal representation of the ASN). Any ASN above the 2-octet range is represented in asdot+ notation which breaks an ASN into two 16-bit values separated by a dot. For example, AS65535 will be represented by the decimal number "65535" while AS65536 will be represented as "1.0".

The ORIGIN AS field MUST be present.

6.3. SRO RR Examples

The following example shows an SRO RR authorizing AS14041 as the origin for CIDR address block 129.82.0.0/16 in the reverse DNS.

```
82.129.in-addr.arpa.  86400  IN   SRO 12145
```

The next example shows two separate origins to be authorized for a prefix. This example also illustrates the use of the asdot notation.

```
82.129.in-addr.arpa.  86400  IN   SRO 12145
                      86400  IN   SRO 3.1858
```

7. Discussion and Related Work

This work is not the first to propose entering routing data in the Reverse DNS and there are also many other proposed approaches for publishing routing data. We first review some of the past work and then discusses the differences presented in this approach.

7.1. Prior Work on CIDR names for Routing

Over a decade ago, [I-D.bates-bgp4-nlri-orig-verif] proposed to use the reverse DNS to verify the origin AS associated with a prefix. This requires both a naming convention for converting the name into a prefix and additional resource record types for storing origin information, along with recommendations on their use. More recently [I-D.donnerhacke-sidr-bgp-verification-dnssec] including links to IRR data and also includes the notion of policy in adjacency, but this approach also introduces a new reverse DNS tree under "BGP.ARPA." CNAME and DNAME records must be used in publishing the data.

Our approach differs in several respects. We rely on the existing reverse DNS tree without creating a new hierarchy such as "BGP.ARPA.". We exploit the naming convention in [I-D.gersch-dnsop-revDNS-CIDR] so one does not need to introduce CNAME or DNAME records (though an operator could choose to do so if so desired). We assume optional participation and introduce the concept of an RLOCK resource record to indicate participation. We currently limit our approach to detecting false sub-prefix and false origin route announcements. Extensions to include links to other databases such as IRR can be achieved in combination with or in lieu of an SRO record and further path validation can be included, but the scope of this document is intentionally limited, both for clarity and to match actual implementation. Finally, we separate the publishing technique which is specified in this document from the variety of ways in which one may make use of the data, recognizing that different operators will make different choices on how to make use of the data.

7.2. RPKI

A great deal of work has been done in the sidr working group on Resource Public Key Infrastructure [RFC6480][RFC6481][RFC6482][RFC6483].

RPKI, also known as Resource Certification, is a specialized public key infrastructure (PKI) framework designed to secure Border Gateway Protocol (BGP). RPKI provides a way to connect Internet number resource information (such as Autonomous System numbers and IP Addresses) to a trust anchor. The certificate structure mirrors the way in which Internet number resources are distributed. That is, resources are initially distributed by the IANA to the Regional Internet Registries (RIRs), who in turn distribute them to Local Internet Registries (LIRs), who then distribute the resources to their customers. RPKI can be used by the legitimate holders of the resources to control the operation of Internet routing protocols to prevent route hijacking and other attacks. [cited from Wikipedia].

The publication of BGP route origin information in the reverse-DNS is a complementary technique to RPKI. While there is some overlap in the techniques, there are also different goals for the reverse-DNS.

The Reverse-DNS publication method uses DNSSEC as its base trust model, not a chain of certificates. If an organization has a DNSSEC-signed delegation for a reverse-DNS address block, that organization is the legitimate owner and may place SRO and RLOCK statements in their zone without the interaction of any other organization. If an address block is sold or transferred, either the RIR (Regional Internet Registry) will change its signed delegation records to reflect the change, or the organization itself may independently implement a signed sub-delegation.

8. Security Considerations

Applications that query the DNS for SRO and RLOCK records MUST request them from DNSSEC-enabled servers and have the DO bit set. Responses that are returned MUST be checked to verify that the D bit is set indicating that the responses have been validated. Otherwise the response should be ignored.

The absence of DNSSEC or the inability to contact any nameservers MUST indicate the route is viable.

9. IANA Considerations

RRType numbers need to be assigned for the SRO and RLOCK records. The current testbed temporarily substitutes TYPE65400 for the RLOCK record and TYPE65401 for the SRO record.

10. Acknowledgments

We would like to thank Danny McPherson for his comments and suggestions. In addition, this document was aided via numerous discussions at NANOG, IETF and private meetings with ISPs, telecomm carriers, and research organizations too numerous to mention by name. Thanks to all for your comments and advice.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.bates-bgp4-nlri-orig-verif] Bates, T., Bush, R., Li, T., and Y. Rekhter, "DNS-based NLRI origin AS verification in BGP", draft-bates-bgp4-nlri-orig-verif-00 (work in progress), January 1998.

[I-D.donnerhacke-sidr-bgp-verification-dnssec] Donnerhacke, L. and W. Wijngaards, "DNSSEC protected routing announcements for BGP", draft-donnerhacke-sidr-bgp-verification-dnssec-04 (work in progress), May 2008.

[I-D.gersch-dnsop-revDNS-CIDR] Gersch, J. and D. Massey, "Reverse DNS Naming Convention for CIDR Address Blocks", draft-gersch-dnsop-revDNS-CIDR-00 (work in progress), February 2012.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, February 2012.

Appendix A. Examples

A.1. Example 1

This example shows data entered for the prefix 129.82.0.0/16. The prefix owner has authorized the announcement of 129.82.0.0/16 and the four /18’s at 129.82.0.0/18, 129.82.64.0/18, 129.82.128.0/18, and 129.82.192.0/18. All the prefixes originate from AS12145.

Finally, the example shows a record for a 129.82.177/24 so that the parent zone can manage this for the child zone at 177.82.129.in-addr.arpa. Any entry in the child zone would override the data stored at the parent.

Note: this data is directly cut and paste from actual deployment. TYPE 65400 is being used for RLOCK and TYPE 65401 for SRO records. This draft requests IANA to assign numbers for RLOCK and SRO, the values here are purely for illustrative purposes.

```
$TTL 3600
$ORIGIN 82.129.in-addr.arpa.

@    IN    SOA     rush.colostate.edu.  dnsadmin.colostate.edu. (
                        2012021300      ; serial number
                        900             ; refresh, 15 minutes
                        600             ; update retry, 10 minutes
                        86400           ; expiry, 1 day
                        3600            ; minimum, 1 hour
                       )

     IN    NS      dns1.colostate.edu.
     IN    NS      dns2.colostate.edu.

@                  IN   TYPE65400 \# 0
;                  RLOCK    OPT-IN; deny all route announcements
;                           except those authorized

@                  IN   TYPE65401 \# 4 00002f71
; 129.82.0.0/16         SRO  12145

0.0.m              IN   TYPE65401 \# 4 00002f71
; 129.82.0.0/18         SRO 12145

1.0.m              IN   TYPE65401 \# 4 00002f71
; 129.82.64.0/18        SRO 12145

0.1.m              IN   TYPE65401 \# 4 00002f71
; 129.82.128.0/18       SRO 12145

1.1.m              IN   TYPE65401 \# 4 00002f71
; 129.82.192.0/18       SRO 12145

1.0.0.0.1.1.0.1.m  IN   TYPE65401 \# 4 00004070
; 129.82.177.0/24       SRO 16496

;  delegations required for 256 /24 zones which contain PTR records

1   IN  NS  dns1.colostate.edu.
    IN  NS  dns2.colostate.edu.
2   IN  NS  dns1.colostate.edu.
    IN  NS  dns2.colostate.edu.

;  continuation to 255 is left out for the sake of brevity
```

A.2. Example 2

This example shows data entered for the prefix 216.17.128.0/17. The prefix owner has authorized the announcement of 216.17.128.0/17. The prefix originates from AS6582.

```
        1.m.17.216.in-addr.arpa  NS   ns.frii.net
```

This delegation refers to the new /17 zone and the domain name is not in conflict with any of the pre-existing /24 zones at IN-ADDR.ARPA. This delegation is to be placed at the IN-ADDR.ARPA zone.

```
 $TTL 3600
 $ORIGIN 1.m.17.216.in-addr.arpa.

 @    IN   SOA     ns1.frii.net.  hostmaster.frii.net. (
                         2012021300      ; serial number
                         14400           ; refresh, 4 hours
                         3600            ; update retry, 1 hour
                         604800          ; expiry, 7 days
                         600             ; minimum, 10 minutes
                       )

      IN   NS      ns1.frii.net.
      IN   NS      ns2.frii.net.

 $ORIGIN 17.216.in-addr.arpa.

 1.m               IN   TYPE65400 \# 0
 ;                      RLOCK    OPT-IN; deny all route announcements
 ;                               except those authorized

 1.m               IN   TYPE65401 \# 4 000019b6
 ; 216.17.128.0/17      SRO 6582

 ; no other delegations or PTR records are needed in this zone file
 ; since the /24 delegations are at ARIN at xxx.17.216.IN-ADDR.ARPA
```

Authors' Addresses

Joe Gersch
Secure64 SW Corp
Fort Collins, CO
US

Email: joe.gersch@secure64.com

Dan Massey
Colorado State University
Fort Collins, CO
US

Email: massey@cs.colostate.edu

Eric Osterweil
Verisign
Reston, VA
US

Email: eosterweil@verisign.com

Lixia Zhang
UCLA
Los Angeles, CA
US

Email: lixia@cs.ucla.edu

Network Working Group R. Bush
Internet-Draft Internet Initiative Japan
Intended status: BCP March 13, 2012
Expires: September 14, 2012

BGPsec Operational Considerations
draft-ietf-sidr-bgpsec-ops-04

Abstract

Deployment of the BGPsec architecture and protocols has many operational considerations. This document attempts to collect and present them. It is expected to evolve as BGPsec is formalized and initially deployed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of this Memo

This Internet-Draft will expire on September 14, 2012.

Copyright Notice

Table of Contents

1. Introduction

BGPsec is a new protocol with many operational considerations. It is expected to be deployed incrementally over a number of years. As core BGPsec-capable routers may require large memory and/or modern CPUs, it is thought that origin validation based on the RPKI will occur over the next one to three years and that BGPsec will start to deploy late in that window.

BGPsec relies on widespread propagation of the Resource Public Key Infrastructure (RPKI) [RFC6480]. How the RPKI is distributed and maintained globally and within an operator's infrastructure may be different for BGPsec than for origin validation.

BGPsec need be spoken only by an AS's eBGP speaking, AKA border, routers, and is designed so that it can be used to protect announcements which are originated by small edge routers. This has special operational considerations.

Different prefixes have different timing and replay protection considerations.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], BGPsec, [I-D.lepinski-bgpsec-overview], the RPKI, see [RFC6480], the RPKI Repository Structure, see [RFC6481], and ROAs, see [RFC6482].

3. RPKI Distribution and Maintenance

All non-ROA considerations in the section on RPKI Distribution and Maintenance of [I-D.ietf-sidr-origin-ops] apply.

4. AS/Router Certificates

As described in [I-D.ymbk-bgpsec-rtr-rekeying] routers MAY be capable of generating their own public/private key-pairs and having their certificates signed and published in the RPKI by the RPKI CA system, and/or MAY be given public/private key-pairs by the operator.

A site/operator MAY use a single certificate/key in all their routers, one certificate/key per router, or any granularity in between.

A large operator, concerned that a compromise of one router's key would make other routers vulnerable, MAY accept a more complex certificate/key distribution burden to reduce this exposure.

On the other extreme, an edge site with one or two routers MAY use a single certificate/key.

5. Within a Network

BGPsec is spoken by edge routers in a network, those which border other networks/ASs.

In a fully BGPsec enabled AS, Route Reflectors MUST have BGPsec enabled if and only if there are eBGP speakers in their client cone, i.e. an RR client or the transitive closure of their customers' customers' customers' ....

A BGPsec capable router MAY use the data it receives to influence local policy within its network, see Section 7. In deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy BGPsec capable border routers.

eBGP speakers which face more critical peers or up/downstreams would be candidates for the earliest deployment. Both securing one's own announcements and validating received announcements should be considered in partial deployment.

On the other hand, an operator wanting to monitor router loading, shifts in traffic, etc. will want to deploy incrementally while watching those and similar effects.

As they are not signed, an eBGP listener SHOULD NOT strongly trust unsigned markings such as communities received across a trust boundary.

6. Considerations for Edge Sites

An edge site which does not provide transit and trusts its upstream(s) SHOULD only originate a signed prefix announcement and need not validate received announcements.

BGPsec protocol capability negotiation provides for a speaker signing the data it sends but being unable to accept signed data. Thus a smallish edge router may hold only its own signing key(s) and sign it's announcement but not receive signed announcements and therefore not need to deal with the majority of the RPKI. Thus such routers

CPU, RAM, and crypto needs are trivial and additional hardware should not be needed.

As the vast majority (84%) of ASs are stubs, and they announce the majority of prefixes, this allows for simpler and less expensive incremental deployment. It may also mean that edge sites concerned with routing security will be attracted to upstreams which support BGPsec.

7. Routing Policy

Unlike origin validation based on the RPKI, BGPsec marks a received announcement as Valid or Invalid, there is no NotFound state. How this is used in routing is up to the operator’s local policy. See [I-D.ietf-sidr-pfx-validate].

As BGPsec will be rolled out over years and does not allow for intermediate non-signing edge routers, coverage will be spotty for a long time. Hence a normal operator’s policy SHOULD NOT be overly strict, perhaps preferring valid announcements and giving very low preference, but still using, invalid announcements.

A BGPsec speaker validates signed paths at the eBGP edge.

Local policy on the eBGP edge MAY convey the validation state of a BGP signed path through normal local policy mechanisms, e.g. setting a BGP community, or modifying a metric value such as local-preference or MED. Some MAY choose to use the large Local-Pref hammer. Others MAY choose to let AS-Path rule and set their internal metric, which comes after AS-Path in the BGP decision process.

Because of possible RPKI version skew, an AS Path which does not validate at router R0 might validate at R1. Therefore, signed paths that are invalid and yet propagated (because they are chosen as best path) SHOULD have their signatures kept intact and MUST be signed if sent to external BGPsec speakers.

This implies that updates which a speaker judges to be invalid MAY be propagated to iBGP peers. Therefore, unless local policy ensures otherwise, a signed path learned via iBGP MAY be invalid. If needed, the validation state should be signaled by normal local policy mechanisms such as communities or metrics.

On the other hand, local policy on the eBGP edge might preclude iBGP or eBGP announcement of signed AS Paths which are invalid.

A BGPsec speaker receiving a path SHOULD perform origin validation per [I-D.ietf-sidr-pfx-validate].

If it is known that a BGPsec neighbor is not a transparent route server, and the router provides a knob to disallow a received pCount (prepend count, zero for transparent route servers) of zero, that knob SHOULD be applied. Routers should default to this knob disallowing pCount 0.

To prevent exposure of the internals of BGP Confederations [RFC5065], a BGPsec speaker which is a Member-AS of a Confederation MUST NOT sign updates sent to another Member-AS of the same Confederation.

8. Notes

For protection from attacks replaying BGP data on the order of a day or longer old, re-keying routers with new keys (previously) provisioned in the RPKI is sufficient. For one procedure, see [I-D.rogaglia-sidr-bgpsec-rollover]

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no ’fix’ for this, it is the nature of distributed data with distributed caches.

Operators who manage certificates SHOULD have RPKI Ghostbuster Records (see [I-D.ietf-sidr-ghostbusters]), signed indirectly by End Entity certificates, for those certificates on which others’ routing depends for certificate and/or ROA validation.

Operators should be aware of impending algorithm transitions, which will be rare and slow-paced, see see [I-D.ietf-sidr-algorithm-agility]. They should work with their vendors to ensure support for new algorithms.

As a router must evaluate certificates and ROAs which are time dependent, routers’ clocks MUST be correct to a tolerance of approximately an hour.

If a router has reason to believe its clock is seriously incorrect, e.g. it has a time earlier than 2011, it SHOULD NOT attempt to validate incoming updates. It SHOULD defer validation until it believes it is within reasonable time tolerance.

Servers should provide time service, such as [RFC5905], to client routers.

9. Security Considerations

The major security considerations for the BGPsec protocol are described in [I-D.ietf-sidr-bgpsec-protocol].

10. IANA Considerations

This document has no IANA Considerations.

11. References

11.1. Normative References

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol-01 (work in progress), October 2011.

[I-D.ietf-sidr-ghostbusters]
Bush, R., "The RPKI Ghostbusters Record", draft-ietf-sidr-ghostbusters-16 (work in progress), December 2011.

[I-D.ietf-sidr-origin-ops]
Bush, R., "RPKI-Based Origin Validation Operation", draft-ietf-sidr-origin-ops-15 (work in progress), March 2012.

[I-D.lepinski-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPSEC", draft-lepinski-bgpsec-overview-00 (work in progress), March 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

11.2. Informative References

[I-D.ietf-sidr-algorithm-agility]
Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for RPKI.", draft-ietf-sidr-algorithm-agility-05 (work in progress), January 2012.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-03 (work in progress), October 2011.

[I-D.rogaglia-sidr-bgpsec-rollover]
Gagliano, R., Patel, K., and B. Weis, "BGPSEC router key roll-over as an alternative to beaconing", draft-rogaglia-sidr-bgpsec-rollover-00 (work in progress), March 2012.

[I-D.ymbk-bgpsec-rtr-rekeying]
Turner, S., Patel, K., and R. Bush, "Router Keying for BGPsec", draft-ymbk-bgpsec-rtr-rekeying-00 (work in progress), March 2012.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, August 2007.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: July 9, 2017

R. Bush
Internet Initiative Japan
January 5, 2017

BGPsec Operational Considerations
draft-ietf-sidr-bgpsec-ops-16

Abstract

Deployment of the BGPsec architecture and protocols has many operational considerations. This document attempts to collect and present the most critical and universal. It is expected to evolve as BGPsec is formalized and initially deployed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

This Internet-Draft will expire on July 9, 2017.

Copyright Notice

Table of Contents

1. Introduction

Origin Validation based on the Resource Public Key Infrastructure (RPKI), [RFC6811], is in its early phases. As BGPsec, [I-D.ietf-sidr-bgpsec-protocol] may require larger memory and/or more modern CPUs, it expected to be deployed incrementally over a longer time span. BGPsec is a new protocol with many operational considerations which this document attempts to describe. As with most operational practices, this document will likely evolve.

BGPsec relies on widespread propagation of the RPKI [RFC6480]. How the RPKI is distributed and maintained globally and within an operator’s infrastructure may be different for BGPsec than for origin validation.

BGPsec needs to be spoken only by an AS’s eBGP-speaking border routers. It is designed so that it can be used to protect announcements which are originated by resource constrained edge routers. This has special operational considerations, see Section 6.

Different prefixes may have different timing and replay protection considerations.

2. Suggested Reading

It is assumed that the reader understands BGP, see [RFC4271], BGPsec, [I-D.ietf-sidr-bgpsec-protocol], the RPKI, see [RFC6480], the RPKI Repository Structure, see [RFC6481], and Route Origin Authorizations (ROAs), see [RFC6482].

3. RPKI Distribution and Maintenance

The considerations for RPKI objects (Certificates, Certificate Revocation Lists (CRLs), manifests, Ghostbusters Records [RFC6481]), Trust Anchor Locators (TALs) [RFC7730], cache behaviours of synchronisation and validation from the section on RPKI Distribution and Maintenance of [RFC7115] apply. Specific considerations relating to ROA objects do not apply to this document.

4. AS/Router Certificates

As described in [I-D.ietf-sidr-rtr-keying] BGPsec-speaking routers are capable of generating their own public/private key-pairs and having their certificates signed and published in the RPKI by the RPKI CA system, and/or are given public/private key-pairs by the operator.

A site/operator may use a single certificate/key in all their routers, one certificate/key per router, or any granularity in between.

A large operator, concerned that a compromise of one router’s key would make other routers vulnerable, may deploy a more complex certificate/key distribution burden to reduce this exposure.

At the other end of the spectrum, an edge site with one or two routers may choose to use a single certificate/key.

In anticipation of possible key compromise, a prudent operator SHOULD pre-provision each router’s ’next’ key in the RPKI so there is no propagation delay for provisioning the new key.

5. Within a Network

BGPsec is spoken by edge routers in a network, those which border other networks/ASs.

In an AS where edge routers speak BGPsec and therefore inject BGPsec paths into the iBGP, Route Reflectors MUST have BGPsec enabled if and only if there are eBGP speakers in their client cone, i.e. an RR client or the transitive closure of a client’s customers.

A BGPsec capable router MAY use the data it receives to influence local policy within its network, see Section 7. In deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy BGPsec enabled border routers.

eBGP speakers which face more critical peers or up/downstreams would be candidates for early deployment. Both securing one's own announcements and validating received announcements should be considered in partial deployment.

An operator should be aware that BGPsec, as any other policy change, can cause traffic shifts in their network. And, as with normal policy shift practice, a prudent operator has tools and methods to predict, measure, modify, etc.

On the other hand, an operator wanting to monitor router loading, shifts in traffic, etc. might deploy incrementally while watching those and similar effects.

BGPsec does not sign over communities, so they are not formally trustable. Additionally, outsourcing verification is not prudent security practice. Therefore an eBGP listener SHOULD NOT strongly trust unsigned security signaling, such as communities, received across a trust boundary.

6. Considerations for Edge Sites

An edge site which does not provide transit and trusts its upstream(s) may only originate a signed prefix announcement and not validate received announcements.

An Operator might need to use hardware with limited resources. In such cases, BGPsec protocol capability negotiation allows for a resource constrained edge router to hold only its own signing key(s) and sign its announcements, but not receive signed announcements. Therefore, the router would not have to deal with the majority of the RPKI, potentially saving the need for additional hardware.

As the vast majority of ASs are stubs, and they announce the majority of prefixes, this allows for simpler and less expensive incremental deployment. It may also mean that edge sites concerned with routing security will be attracted to upstreams which support BGPsec.

7. Routing Policy

Unlike origin validation based on the RPKI, BGPsec marks a received announcement as Valid or Not Valid, there is no explicit NotFound state. In some sense, an unsigned BGP4 path is the equivalent of NotFound. How this is used in routing is up to the operator’s local policy, similar to origin validation as in [RFC6811].

As BGPsec will be rolled out over years and does not allow for intermediate non-signing edge routers, coverage will be spotty for a long time. This presents a dilemma; should a router evaluating an inbound BGPsec_Path as Not Valid be very strict and discard it? On the other hand, it might be the only path to that prefix, and a very low local-preference would cause it to be used and propagated only if there was no alternative. Either choice is reasonable, but we recommend dropping because of the next point.

Operators should be aware that accepting Not Valid announcements, no matter the local preference, will often be the equivalent of treating them as fully Valid. Local preference affects only routes to the same set of destinations. Consider having a Valid announcement from neighbor V for prefix 10.0.0.0/16 and an Not Valid announcement for 10.0.666.0/24 from neighbor I. If local policy on the router is not configured to discard the Not Valid announcement from I, then longest match forwarding will send packets to neighbor I no matter the value of local preference.

Validation of signed paths is usually deployed at the eBGP edge.

Local policy on the eBGP edge MAY convey the validation state of a BGP signed path through normal local policy mechanisms, e.g. setting a BGP community for internal use, or modifying a metric value such as local-preference or multi-exit discriminator (MED). Some may choose to use the large Local-Pref hammer. Others may choose to let AS-Path rule and set their internal metric, which comes after AS-Path in the BGP decision process.

As the mildly stochastic timing of RPKI propagation may cause version skew across routers, an AS Path which does not validate at router R0 might validate at R1. Therefore, signed paths that are Not Valid and yet propagated (because they are chosen as best path) MUST NOT have signatures stripped and MUST be signed if sent to external BGPsec speakers.

This implies that updates which a speaker judges to be Not Valid MAY be propagated to iBGP peers. Therefore, unless local policy ensures otherwise, a signed path learned via iBGP may be Not Valid. If needed, the validation state should be signaled by normal local policy mechanisms such as communities or metrics.

On the other hand, local policy on the eBGP edge might preclude iBGP or eBGP announcement of signed AS Paths which are Not Valid.

A BGPsec speaker receiving a path SHOULD perform origin validation per [RFC6811] and [RFC7115].

A route server is usually ’transparent’, i.e. does not insert an AS into the path so as not to increase the AS hop count and thereby affect downstream path choices. But, with BGPsec, a client router R needs to be able to validate paths which are forward signed to R. But the sending router can not generate signatures to all the possible clients. Therefore a BGPsec-aware route server needs to validate the incoming BGPsec_Path, and to forward updates which can be validated by clients which must therefore know the route server’s AS. This implies that the route server creates signatures per client including its own AS in the BGPsec_Path, forward signing to each client AS, see [I-D.ietf-sidr-bgpsec-protocol]. The route server uses pCount of zero to not increase the effective AS hop count, thereby retaining the intent of ’transparency’.

If it is known that a BGPsec neighbor is not a transparent route server, or is otherwise validly using pCount=0 (e,g, see [I-D.ietf-sidr-as-migration]), and the router provides a knob to disallow a received pCount (of zero, that knob SHOULD be applied. Routers should disallow pCount 0 by default.

To prevent exposure of the internals of BGP Confederations [RFC5065], a BGPsec speaker exporting to a non-member removes all intra-confederation Secure_Path segments. Therefore signing within the confederation will not cause external confusion even if non-unique private ASs are used.

8. Notes

For protection from attacks replaying BGP data on the order of a day or longer old, re-keying routers with new keys (previously) provisioned in the RPKI is sufficient. For one approach, see [I-D.ietf-sidr-bgpsec-rollover]

A router that once negotiated (and/or sent) BGPsec should not be expected to always do so.

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix or router than another cache or router. There is no ’fix’ for this, it is the nature of distributed data with distributed caches.

Operators who manage certificates SHOULD have RPKI GhostBuster Records (see [RFC6493]), signed indirectly by End Entity certificates, for those certificates on which others’ routing depends for certificate and/or ROA validation.

Operators should be aware of impending algorithm transitions, which will be rare and slow-paced, see [RFC6916]. They should work with their vendors to ensure support for new algorithms.

As a router must evaluate certificates and ROAs which are time dependent, routers’ clocks MUST be correct to a tolerance of approximately an hour. The common approach is for operators to deploy servers that provide time service, such as [RFC5905], to client routers.

If a router has reason to believe its clock is seriously incorrect, e.g. it has a time earlier than 2011, it SHOULD NOT attempt to validate incoming updates. It SHOULD defer validation until it believes it is within reasonable time tolerance.

9. Security Considerations

This document describes operational considerations for the deployment of BGPsec. The security considerations for BGPsec are described in [I-D.ietf-sidr-bgpsec-protocol].

10. IANA Considerations

This document has no IANA Considerations.

11. Acknowledgments

The author wishes to thank Thomas King, Arnold Nipper, and Alvaro Retana, and the BGPsec design group.

12. References

12.1. Normative References

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol-07 (work in progress), February 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, February 2012.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, January 2013.

[RFC7115] Bush, R., "Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)", BCP 185, RFC 7115, DOI 10.17487/RFC7115, January 2014, <http://www.rfc-editor.org/info/rfc7115>.

[RFC7730] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 7730, DOI 10.17487/RFC7730, January 2016, <http://www.rfc-editor.org/info/rfc7730>.

12.2. Informative References

[I-D.ietf-sidr-as-migration]
George, W. and S. Murphy, "BGPSec Considerations for AS Migration", draft-ietf-sidr-as-migration-06 (work in progress), December 2016.

[I-D.ietf-sidr-bgpsec-rollover]
Gagliano, R., Patel, K., and B. Weis, "BGPSEC router key rollover as an alternative to beaconing", draft-ietf-sidr-bgpsec-rollover-01 (work in progress), October 2012.

[I-D.ietf-sidr-rtr-keying]
Turner, S., Patel, K., and R. Bush, "Router Keying for BGPsec", draft-ietf-sidr-rtr-keying-01 (work in progress), February 2013.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, August 2007.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6916] Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI)", BCP 182, RFC 6916, DOI 10.17487/RFC6916, April 2013, <http://www.rfc-editor.org/info/rfc6916>.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

M. Lepinski, Ed.
BBN

March 12, 2012

BGPSEC Protocol Specification
draft-ietf-sidr-bgpsec-protocol-02

Abstract

This document describes BGPSEC, an extension to the Border Gateway Protocol (BGP) that provides security for the AS-PATH attribute in BGP update messages. BGPSEC is implemented via a new optional non-transitive BGP path attribute that carries a digital signature produced by each autonomous system on the AS-PATH.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

Status of this Memo

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document describes BGPSEC, a mechanism for providing path security for Border Gateway Protocol (BGP) [1] route advertisements. That is, a BGP speaker who receives a valid BGPSEC update has cryptographic assurance that the advertised route has the following two properties:

1. The route was originated by an AS that has been explicitly authorized by the holder of the IP address prefix to originate route advertisements for that prefix.

2. Every AS listed in the AS_Path attribute of the update explicitly authorized the advertisement of the route to the subsequent AS in the AS_Path.

This document specifies a new optional (non-transitive) BGP path attribute, BGPSEC_Path_Signatures. It also describes how a BGPSEC-compliant BGP speaker (referred to hereafter as a BGPSEC speaker) can generate, propagate, and validate BGP update messages containing this attribute to obtain the above assurances.

BGPSEC relies on the Resource Public Key Infrastructure (RPKI) certificates that attest to the allocation of AS number and IP address resources. (For more information on the RPKI, see [7] and the documents referenced therein.) Any BGPSEC speaker who wishes to send BGP update messages to external peers (eBGP) containing the BGPSEC_Path_Signatures must have an RPKI end-entity certificate (as well as the associated private signing key) corresponding to the BGPSEC speaker’s AS number. Note, however, that a BGPSEC speaker does not require such a certificate in order to validate update messages containing the BGPSEC_Path_Signatures attribute.

2. BGPSEC Negotiation

This document defines a new BGP capability [3]that allows a BGP speaker to advertise to its neighbors the ability to send and/or receive BGPSEC update messages (i.e., update messages containing the BGPSEC_Path_Signatures attribute).

This capability has capability code : TBD

The capability length for this capability MUST be set to 5.

The three octets of the capability value are specified as follows.

```
                             Capability Value:

                    0        1        2  3       4 5 6 7
                 +---------------------------------------+
                 | Send | Receive | Reserved  |  Version |
                 +---------------------------------------+
                 |              AFI                      |
                 +---------------------------------------+
                 |                                       |
                 +---------------------------------------+
                 |            Reserved                   |
                 +---------------------------------------+
                 |              SAFI                     |
                 +---------------------------------------+
```

The high order bit (bit 0) of the first octet is set to 1 to indicate that the sender is able to send BGPSEC update messages, and is set to zero otherwise. The next highest order bit (bit 1) of this octet is set to 1 to indicate that the sender is able to receive BGPSEC update messages, and is set to zero otherwise. The next two bits of the capability value (bits 2 and 3) are reserved for future use. These reserved bits should be set to zero by the sender and ignored by the receiver.

The four low order bits (4, 5, 6 and 7) of the first octet indicate the version of BGPSEC for which the BGP speaker is advertising support. This document defines only BGPSEC version 0 (all four bits set to zero). Other versions of BGPSEC may be defined in future documents. A BGPSEC speaker MAY advertise support for multiple versions of BGPSEC by including multiple versions of the BGPSEC capability in its BGP OPEN message.

If there does not exist at least one version of BGPSEC that is supported by both peers in a BGP session, then the use of BGPSEC has not been negotiated. (That is, in such a case, messages containing the BGPSEC_Path_Signatures MUST NOT be sent.)

If version 0 is the only version of BGPSEC for which both peers (in a BGP session) advertise support, then the use of BGPSEC has been negotiated and the BGPSEC peers MUST adhere to the specification of BGPSEC provided in this document. (If there are multiple versions of BGPSEC which are supported by both peers, then the behavior of those peers is outside the scope of this document.)

The second and third octets contain the 16-bit Address Family Identifier (AFI) which indicates the address family for which the BGPSEC speaker is advertising support for BGPSEC. This document only specifies BGPSEC for use with two address families, IPv4 and IPv6, AFI values 1 and 2 respectively. BGPSEC for use with other address families may be specified in future documents.

The fourth octet in the capability is reserved. It is anticipated that this octet will not be used until such a time as the reserved octet in the Multi-protocol extensions capability advertisement [2] is specified for use. The reserved octet should be set to zero by the sender and ignored by the receiver.

The fifth octet in the capability contains the 8-bit Subsequent Address Family Identifier (SAFI). This value is encoded as in the BGP multiprotocol extensions [2].

Note that if the BGPSEC speaker wishes to use BGPSEC with two different address families (i.e., IPv4 and IPv6) over the same BGP session, then the speaker must include two instances of this capability (one for each address family) in the BGP OPEN message. A BGPSEC speaker SHOULD NOT advertise the capability of BGPSEC support for any <AFI, SAFI> combination unless it has also includes the multiprotocol extension capability for the same <AFI, SAFI> combination [2].

By indicating support for receiving BGPSEC update messages, a BGP speaker is, in particular, indicating that the following are true:

- The BGP speaker understands the BGPSEC_Path_Signatures attribute (see Section 3).

- The BGP speaker supports 4-byte AS numbers (see RFC 4893).

Note that BGPSEC update messages can be quite large, therefore any BGPSEC speaker announcing the capability to receive BGPSEC messages SHOULD also announce support for the capability to receive BGP extended messages [5].

A BGP speaker MUST NOT send an update message containing the BGPSEC_Path_Signatures attribute within a given BGP session unless both of the following are true:

- The BGP speaker indicated support for sending BGPSEC update messages in its open message.

- The peer of the BGP speaker indicated support for receiving BGPSEC update messages in its open message.

3. The BGPSEC_Path_Signatures Attribute

The BGPSEC_Path_Signatures attribute is a new optional (non-transitive) BGP path attribute.

This document registers a new attribute type code for this attribute : TBD

The BGPSEC_Path_Signatures attribute has the following structure:

```
              BGPSEC_Path_Signatures Attribute
+---------------------------------------------------------+
| Flags Octet                     (1 octet)               |
+---------------------------------------------------------+
| Algorithm Suite Identifier 1    (1 octet)               |
+---------------------------------------------------------+
| Algorithm Suite Identifier 2    (1 octet)               |
+---------------------------------------------------------+
| Reserved                       (8 octets)               |
+---------------------------------------------------------+
| Sequence of Signature-Segments (variable)               |
+---------------------------------------------------------+
```

The flags octet is an unsigned octet that contains flags to aid in receiver processing.

```
     Flags Octet in Path_Signatures Attribute

            0              1  2  3  4  5  6  7
     +-------------------------------------------+
     | Two Algorithms  |        Reserved         |
     +-------------------------------------------+
```

The first bit in the Flags octet is set to zero in the common case that each Signature-Segment contains a single signature. The first bit of the Flags octet is set to one in the case that each Signature-Segment contains two signatures, produced by two different algorithm suites. (Note that this second case is necessary to support a transition between two algorithm suites, see Section 8.) The remaining 7 bits of the Flags octet are reserved for future use. These bits should be set to zero by the sender and ignored by the receiver.

Algorithm Suite Identifier 1 contains a one-octet identifier specifying the digest algorithm and digital signature algorithm used to produce the first signature in each Signature-Segment. An IANA registry of algorithm identifiers for use in BGPSEC is created in the BGPSEC algorithms document[10].

Algorithm Suite Identifier 2 contains a one-octet identifier specifying the digest algorithm and digital signature algorithm used to produce the second signature in each Signature-Segment. This field is ignored by the receiver if the first bit in the Flags octet is set to zero (indicating that only one signature algorithm is used in this BGPSEC update). An IANA registry of algorithm identifiers for use in BGPSEC is created in the BGPSEC algorithms document[10].

There are eight octets reserved for future use. These octets are digitally signed (see Section 4 below).

EDITOR’S NOTE: In a previous version of this document there was an Expire Time that was used to provide protection against replay of old (stale) digital signatures or failure to propagate a withdrawal message. This mechanism was removed from the current version of the document. Please see the SIDR mailing list for discussions related to protection against replay attacks. Depending on the result of discussions within the SIDR working group this reserved field could at some future point be used to re-introduce Expire Time, or some other octets used in a future replay protection mechanism.

The BGPSEC_Path_Signatures attribute contains one Signature-Segment for each AS along the path of the route advertisement in this update message. (For a detailed explanation of how an AS processes a BGPSEC update message and adds a new Signature_Segment, see Section 4.) A Signature-Segment has the following structure:

Signature Segments

```
+-------------------------------------------- +
| AS Number                       (4 octets)  |
+---------------------------------------------+
| pCount                           (1 octet)  |
+---------------------------------------------+
| Subject Key Identifier 1 Length  (1 octet)  |
+---------------------------------------------+
| Subject Key Identifier 1        (variable)  |
+---------------------------------------------+
| Signature 1 Length               (1 octet)  |
+---------------------------------------------+
| Signature 1                     (variable)  |
+---------------------------------------------+
| Subject Key Identifier 2 Length  (1 octet)  |
+---------------------------------------------+
| Subject Key Identifier 2        (variable)  |
+---------------------------------------------+
| Signature Length 2               (1 octet)  |
+---------------------------------------------+
| Signature 2                     (variable)  |
+---------------------------------------------+
```

The AS Number is the Autonomous System Number of the BGPSEC speaker that produced the digital signature(s) in this Signature Segment.

The pCount field contains an unsigned integer indicating the number of repetitions of the associated autonomous system number that the signature covers. This field enables a BGPSEC speaker to mimic the semantics of adding multiple copies of their AS to the AS-PATH without requiring the speaker to generate multiple signatures.

The Subject Key Identifier 1 Length field contains the size (in octets) of the value in the Subject Key Identifier 1 field of the Signature-Segment. The Subject Key Identifier 1 field contains the value in the Subject Key Identifier extension of the RPKI end-entity certificate that is used to verify the first signature in the Signature-Segment (see Section 5 for details on validity of BGPSEC update messages).

The Signature 1 Length field contains the size (in octets) of the value in the Signature 1 field. The Signature 1 field contains a digital signature that protects the NLRI and the BGPSEC_Path_Signatures attribute (see Sections 4 and 5 for details on generating and verifying this signature, respectively).

The Subject Key Identifier 2 Length field contains the size (in octets) of the value in the Subject Key Identifier 2 field of the Signature-Segment. This length field SHOULD be zero if the first bit in the Flags octet is zero (indicating that only one algorithm suite is being used to generate signatures for this update message). The Subject Key Identifier 2 field contains the value in the Subject Key Identifier extension of the RPKI end-entity certificate that is used to verify the second signature in the Signature-Segment (see Section 5 for details on validity of BGPSEC update messages). This field is ignored by the receiver when the first bit in the Flags octet is zero (indicating that only one algorithm suite is being used to generate signatures for this update message).

The Signature 2 Length field contains the size (in octets) of the value in the Signature 2 field. This length field SHOULD be zero if the first bit in the Flags octet is zero (indicating that only one algorithm suite is being used to generate signatures for this update message). The Signature 2 field contains a digital signature that protects the NLRI and the BGPSEC_Path_Signatures attribute (see Sections 4 and 5 for details on generating and verifying this signature, respectively). This field is ignored by the receiver when the first bit in the Flags octet is zero (indicating that only one algorithm suite is being used to generate signatures for this update message).

4. Generating a BGPSEC Update

Sections 4.1 and 4.2 cover two cases in which a BGPSEC speaker may generate an update message containing the BGPSEC_Path_Signatures attribute. The first case is that in which the BGPSEC speaker originates a new route advertisement (Section 4.1). That is, the BGPSEC speaker is constructing an update message in which the only AS to appear in the AS_PATH attribute is the speaker’s own AS (normally appears once but may appear multiple times if AS prepending is applied). The second case is that in which the BGPSEC speaker receives a route advertisement from a peer and then decides to propagate the route advertisement to an external (eBGP) peer (Section 4.2). That is, the BGPSEC speaker has received a BGPSEC update message and is constructing a new update message for the same NLRI in which the AS_PATH attribute will contain AS number(s) other than the speaker’s own AS.

In the remaining case where the BGPSEC speaker is sending the update message to an internal (iBGP) peer, the BGPSEC speaker populates the BGPSEC_Path_Signatures attribute by copying the BGPSEC_Path_Signatures attribute from the received update message. That is, the BGPSEC_Path_Signatures attribute is copied verbatim.

Note that in the case that a BGPSEC speaker chooses to forward to an iBGP peer a BGPSEC update message that has not been successfully validated (see Section 5), the BGPSEC_Path_Signatures attribute SHOULD NOT be removed. (See Section 7 for the security ramifications of removing BGPSEC signatures.)

The information protected by the signature on a BGPSEC update message includes the AS number of the peer to whom the update message is being sent. Therefore, if a BGPSEC speaker wishes to send a BGPSEC update to multiple BGP peers, it MUST generate a separate BGPSEC update message for each unique peer AS to which the update message is sent.

A BGPSEC update message MUST advertise a route to only a single NLRI. This is because a BGPSEC speaker receiving an update message with multiple NLRI is unable to construct a valid BGPSEC update message (i.e., valid path signatures) containing a subset of the NLRI in the received update. If a BGPSEC speaker wishes to advertise routes to multiple NLRI, then it MUST generate a separate BGPSEC update message for each NLRI.

Note that in order to create or add a new signature to a BGPSEC update message with a given algorithm suite, the BGPSEC speaker must possess a private key suitable for generating signatures for this algorithm suite. Additionally, this private key must correspond to the public key in a valid Resource PKI end-entity certificate whose AS number resource extension includes the BGPSEC speaker’s AS number [11]. Note also new signatures are only added to a BGPSEC update message when a BGPSEC speaker is generating an update message to send to an external peer (i.e., when the AS number of the peer is not equal to the BGPSEC speaker’s own AS number). Therefore, a BGPSEC speaker who only sends BGPSEC update messages to peers within its own AS, it does not need to possess any private signature keys.

4.1. Originating a New BGPSEC Update

In an update message that originates a new route advertisement (i.e., an update whose AS_Path contains a single AS number), a BGPSEC speaker will use only a single algorithm suite. That is, the BGPSEC speaker will set the Two_Algorithms flag to 0 in the BGPSEC_Path_Signatures attribute and include only a single signature in the Signature-Segment (setting the Signature 2 Length and Subject Key Identifier 2 Lengths to zero). However, to ensure backwards compatibility during a period of transition from a ’current’ algorithm suite to a ’new’ algorithm suite, it will be necessary to originate update messages containing both the ’current’ and the ’new’ algorithm suites (see Section 6.1). In such a case the BGPSEC speaker will set the Two_Algorithms flag to 1 in the

BGPSEC_Path_Signatures attribute and include two separate digital signatures (one for each algorithm suite). For the remainder of this section we describe the common case where the Two_Algorithms flag is set to one. However, the construction of the second signature is completely analogous (the only change is the replacement of 1 by 2 in the field names corresponding to the second signature).

The Resource PKI enables the legitimate holder of IP address prefix(es) to issue a signed object, called a Route Origination Authorization (ROA), that authorizes a given AS to originate routes to a given set of prefixes (see [6]). Note that validation of a BGPSEC update message will fail (i.e., the validation algorithm, specified in Section 5.1, returns ’Not Good’) unless there exists a valid ROA authorizing the first AS in the AS PATH attribute to originate routes to the prefix being advertised. Therefore, a BGPSEC speaker SHOULD NOT originate a BGPSEC update advertising a route for a given prefix unless a ROA has previously been created (and published in the repository system) that authorizing the BGPSEC speaker’s AS to originate routes to this prefix.

EDITOR’S NOTE: In a previous version of this document there was a description here of a mechanism that used that used periodic repetition of update messages (aka "beaconing") to protect against replay of old (stale) digital signatures or failure to propagate a withdrawal message. This mechanism was removed from the current version of the document. Please see the SIDR mailing list for discussions related to protection against replay attacks. Depending on the result of discussions within the SIDR working group a mechanism for protection against replay of digital signatures may be re-introduced into BGPSEC in the future.

When originating a new route advertisement, the BGPSEC_Path_Signatures attribute MUST contain a single Signature-Segment. The following describes how the BGPSEC speaker populates the fields of the Signature-Segment (see Section 3 for more information on the syntax of the Signature-Segment).

The AS field is set to the AS number of the BGPSEC speaker. That is, the AS number that the BGPSEC speaker advertised in the Open message of the current BGP session.

The pCount field is typically set to the value 1. However, a BGPSEC speaker may set the pCount field to a value greater than 1. Setting the pCount field to a value greater than one has the same semantics as repeating an AS number multiple times in the AS_PATH of a non-BGPSEC update message (e.g., for traffic engineering purposes). Setting the pCount field to a value greater than one permits this repetition without requiring a separate digital signature for each repetition.

The Subject Key Identifier 1 field (see Section 3) is populated with the identifier contained in the Subject Key Identifier extension of the RPKI end-entity certificate (containing keys suitable for use with Algorithm Suite 1) used by the BGPSEC speaker. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Subject Key Identifier 1 Length field is populated with the length (in octets) of the Subject Key Identifier 1 field.

The Signature 1 field contains a digital signature that binds the NLRI, AS_Path attribute and BGPSEC_Path_Signatures attribute to the RPKI end-entity certificate used by the BGPSEC speaker. The digital signature is computed as follows:

- o Construct a sequence of octets by concatenating the Target AS Number, AS Number (from the Signature_Segment), pCount, Algorithm Suite Identifier 1, Reserved field of the BGPSEC_Path_Signatures attribute and NLRI. The Target AS Number is the AS to whom the BGPSEC speaker intends to send the update message. (Note that the Target AS number is the AS number announced by the peer in the OPEN message of the BGP session within which the update is sent.)

Sequence of Octets to be Signed

Target AS Number	(4 octets)
AS Number	(4 octets)
pCount	(1 octet)
Algorithm Suite Identifier 1	(1 octet)
Expire Time	(8 octets)
NLRI Length	(1 octet)
NLRI Prefix	(variable)

- o Apply to this octet sequence the digest algorithm (for Algorithm Suite 1) to obtain a digest value.

- o Apply to this digest value the signature algorithm, (for Algorithm Suite 1) to obtain the digital signature. Then populate the Signature 1 field with this digital signature.

The Signature 1 Length field is populated with the length (in octets) of the Signature 1 field.

4.2. Propagating a Route Advertisement

When a BGPSEC speaker receives a BGPSEC update message containing a BGPSEC_Path_Signatures algorithm (with one or more signatures) from a (internal or external) peer, it may choose to propagate the route advertisement by sending to its (internal or external) peers by creating a new BGPSEC advertisement for the same prefix.

A BGPSEC speaker MUST NOT generate an update message containing the BGPSEC_Path_Signatures attribute unless it has selected, as the best route to the given prefix, a route that it received in an update message containing the BGPSEC_Path_Signatures attribute. In particular, this means that whenever a BGPSEC speaker generates an update message with a BGPSEC_Path_Signatures attribute that it will possess a received update message for the same prefix that also contains a BGPSEC_Path_Signatures attribute.

Additionally, whenever a BGPSEC speaker selects as the best route to a given prefix a route that it received in an update message containing the BGPSEC_Path_Signatures attribute, it is RECOMMENDED that if the BGPSEC speaker chooses to propagate the route that it generate an update message containing the BGPSEC_Path_Signatures attribute. However, a BGPSEC speaker MAY propagate a route advertisement by generating a (non-BGPSEC) update message that does not contain the BGPSEC_Path_Signatures attribute. Note that if a BGPSEC speaker receives a route advertisement containing the BGPSEC_Path_Signatures attribute and chooses for any reason (e.g., its peer is a non-BGPSEC speaker) to propagate the route advertisement as a non-BGPSEC update message without the BGPSEC_Path_Signatures attribute, then it MUST follow the instructions in Section 4.2.1.

The Subject Key Identifier 1 field (see Section 3) is populated with the identifier contained in the Subject Key Identifier extension of the RPKI end-entity certificate (containing keys suitable for use with Algorithm Suite 1) used by the BGPSEC speaker. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Subject Key Identifier 1 Length field is populated with the length (in octets) of the Subject Key Identifier 1 field.

Note that removing BGPSEC signatures (i.e., propagating a route advertisement without the BGPSEC_Path_Signatures attribute) has significant security ramifications. (See Section 7 for discussion of the security ramifications of removing BGPSEC signatures.) Therefore, when a route advertisement is received via a BGPSEC update message, propagating the route advertisement without the BGPSEC_Path_Signatures attribute is NOT RECOMMENDED. Furthermore, note that when a BGPSEC speaker propagates a route advertisement with the BGPSEC_Path_Signatures attribute it is attesting to the fact that: (1) it received a BGPSEC update message that advertised this route; and (2) it chose this route as its best path to the given prefix. That is, the BGPSEC speaker is not attesting to the validation state of the update message it received. (See Section 7 for more discussion of the security semantics of BGPSEC signatures.)

If the BGPSEC speaker is producing an update message which contains an AS-SET (e.g., the BGPSEC speaker is performing proxy aggregation), then the BGPSEC speaker MUST NOT include the BGPSEC_Path_Signatures attribute. In such a case, the BGPSEC speaker must remove any existing BGPSEC_Path_Signatures in the received advertisement(s) for this prefix and produce a standard (non-BGPSEC) update message.

If the received BGPSEC update message uses two algorithm suites (i.e., the Two_Algorithms flag is set to 1) and the BGPSEC speaker supports both of the corresponding algorithms suites, then the BGPSEC speaker SHOULD generate a new update message that uses both algorithm suites (i.e., set the Two_Algorithms flag to 1). If the received BGPSEC update message that uses two algorithm suites and the BGPSEC speaker does not support the second algorithm suite, then the BGPSEC speaker MUST set the Two_Algorithms flag to 1 and remove the Signature 2 and Subject Key Identifier 2 fields from each Signature-Segment in the BGPSEC_Path_Signatures attribute (and set the corresponding lengths to zero). Note that this case can happen during an algorithm transition when the BGPSEC speaker has not yet been updated to support the new algorithm, see Section 6 for more details. If the BGPSEC speaker does not support the first algorithm suite in a BGPSEC update message, then the BGPSEC speaker MUST NOT propagate the route advertisement with the BGPSEC_Path_Signatures attribute. (Note that if this case occurs, something has gone wrong, as algorithm transitions are designed to never produce this case.)

The Reserved field from the BGPSEC_Path_Signatures attribute is copied directly from the Reserved field in the received update message.

The BGPSEC speaker then creates a new Signature-Segment. This Signature-Segment is prepended to the list of Signature-Segments (placed in the first position) so that the list of Signature-Segments appears in the same order as the corresponding AS numbers in the AS_PATH attribute. The BGPSEC speaker populates the fields of this new Signature-Segment as follows.

The AS field is set to the AS number of the BGPSEC speaker. That is, the AS number that the BGPSEC speaker advertised in the Open message of the current BGP session.

The pCount is typically set to the value 1. A BGPSEC speaker may set the pCount field to a value greater than 1. (See Section 4.1 for a discussion of setting pCount to a value greater than 1.) A route server that participates in the BGP control path, but does not act as a transit AS in the data plane, may choose to set pCount to 0. This option enables the route server to participate in BGPSEC and obtain the associated security guarantees without increasing the effective length of the AS_PATH. (Note that the Signature_Segmenet still contains the AS Number of the route server as this information is necessary for signature verification.) Note that the option of setting pCount to 0 is intended only for use by route servers that desire not to increase the effective AS-PATH length of routes they advertise. The pCount field SHOULD NOT be set to 0 in other circumstances. BGPSEC speakers SHOULD drop incoming update messages with pCount set to zero in cases where the BGPSEC speaker does not expect its peer to set pCount to zero (i.e., cases where the peer is not acting as a route server).

The Subject Key Identifier 1 field (see Section 3) is populated with the identifier contained in the Subject Key Identifier extension of the RPKI end-entity certificate (containing keys suitable for use with Algorithm Suite 1) used by the BGPSEC speaker. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Subject Key Identifier 1 Length field is populated with the length (in octets) of the Subject Key Identifier 1 field.

The Signature 1 field in the new segment contains a digital signature that binds the NLRI, AS_Path attribute and BGPSEC_Path_Signatures attribute to the RPKI end-entity certificate used by the BGPSEC speaker. The digital signature is computed as follows:

o Construct a sequence of octets by concatenating the Signature 1 Length and Signature 1 fields of the most recent Signature-Segment (the one corresponding to AS from whom the BGPSEC speaker’s AS received the announcement) with the pCount field inserted by the signer, and the Target AS (the AS to whom the BGPSEC speaker intends to send the update message). Note that the Target AS number is the AS number announced by the peer in the OPEN message of the BGP session within which the BGPSEC update message is sent.

Sequence of Octets to be Signed

```
      +-------------------------------------------------------------+
      | Most Recent Signature 1 Length Field      (1 octet)         |
      +-------------------------------------------------------------+
      | Most Recent Signature 1 Field             (variable)        |
      +-------------------------------------------------------------+
      | pCount Field of Signer       (1 octet)                      |
      +-------------------------------------------------------------+
      | Target AS Number             (4 octets)                     |
      +-------------------------------------------------------------+
```

- Apply to this octet sequence the digest algorithm (for the algorithm suite of this Signature-List) to obtain a digest value.

- Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature-List) to obtain the digital signature. Then populate the Signature Field with this digital signature.

The Subject Key Identifier 1 Length field is populated with the length (in octets) of the Subject Key Identifier 1 field.

5. Processing a Received BGPSEC Update

Validation of a BGPSEC update messages makes use of data from RPKI certificates and signed Route Origination Authorizations (ROA). In particular, to validate update messages containing the BGPSEC_Path_Signatures attribute, it is necessary that the recipient have access to the following data obtained from valid RPKI certificates and ROAs:

- For each valid RPKI end-entity certificate containing an AS Number extension, the AS Number, Public Key and Subject Key Identifier are required

- For each valid ROA, the AS Number and the list of IP address prefixes

Note that the BGPSEC speaker could perform the validation of RPKI certificates and ROAs on its own and extract the required data, or it could receive the same data from a trusted cache that performs RPKI validation on behalf of (some set of) BGPSEC speakers. (The latter case in analogous to the use of the RPKI-RTR protocol [12] for origin validation.)

To validate a BGPSEC update message containing the

BGPSEC_Path_Signatures attribute, the recipient performs the validation steps specified in Section 5.1. The validation procedure results in one of two states: ’Good’ and ’Not Good’.

It is expected that the output of the validation procedure will be used as an input to BGP route selection. However, BGP route selection and thus the handling of the two validation states is a matter of local policy, and shall be handled using existing local policy mechanisms. It is expected that BGP peers will generally prefer routes received via ’Good’ BGPSEC update messages over routes received via ’Not Good’ BGPSEC update messages as well as routes received via update messages that do not contain the BGPSEC_Path_Signatures attribute. However, BGPSEC specifies no changes to the BGP decision process and leaves to the operator the selection of an appropriate policy mechanism to achieve the operator’s desired results within the BGP decision process.

BGPSEC validation need only be performed at eBGP edge. The validation status of a BGP signed/unsigned update MAY be conveyed via iBGP from an ingress edge router to an egress edge router. Local policy in the AS determines the specific means for conveying the validation status through various pre-existing mechanisms (e.g., modifying an attribute). As discussed in Section 4, when a BGPSEC speaker chooses to forward a (syntactically correct) BGPSEC update message, it SHOULD be forwarded with its BGPSEC_Path_Signatures attribute intact (regardless of the validation state of the update message). Based entirely on local policy settings, an egress router MAY trust the validation status conveyed by an ingress router or it MAY perform its own validation.

EDITOR’S NOTE: Text will be inserted here for dealing with the AS_PATH attribute. Note that the BGPGSEC_Path_Signatures attribute now contains all of the information needed to construct the AS_PATH attribute. Therefore, there seem to be two options. One option the BGPSEC speaker checks the AS_PATH attribute against the information in the BGPSEC_Path_Signatures attribute and returns "Not Good" if the two do not match. The other option is that the BGPSEC speaker discards anything in the AS_PATH attribute and reconstructs the AS_PATH from the data in the BGPSEC_Path_Signatures attribute. I believe that there are no interoperability problems if the choice between these two options is left up to the BGPSEC speaker.

5.1. Validation Algorithm

This section specifies an algorithm for validation of BGPSEC update messages. A conformant implementation MUST include an BGPSEC update validation algorithm that is functionally equivalent to the external behavior of this algorithm.

First, the recipient of a BGPSEC update message performs a check to ensure that the message is properly formed. Specifically, the recipient checks that the BGPSEC_Path_Signatures attribute is properly formed (as specified in Section 3). If the BGPSEC_Path_Signatures attribute is not properly formed, then the recipient should log that an error occurred and drop the update message containing the error.

Second, the BGPSEC speaker verifies that the origin AS is authorized to advertise the prefix in question. To do this, consult the valid ROA data to obtain a list of AS numbers that are associated with the given IP address prefix in the update message. Then locate the last (least recently added) AS number in the AS-Path. If the origin AS in the AS-Path is not in the set of AS numbers associated with the given prefix, then BGPSEC update message is 'Not Good' and the validation algorithm terminates.

Third, the BGPSEC speaker examines the Algorithm Suite identifiers and the Two-Algorithms flag in the BGPSEC_Path_Signatures attribute. If the BGPSEC speaker does not support the first Algorithm Suite, then the BGPSEC speaker MUST treat the update message in the same manner that the BGPSEC speaker would treat an update message that arrived without a BGPSEC_Path_Signatures attribute. (Note that algorithm transitions are designed so that this case will never happen, therefore if this case occurs the BGPSEC speaker SHOULD log an error message.) If the Two-Algorithms flag is set to 1 and the BGPSEC speaker supports only the first algorithm suite then it follows the instructions below to validate the signatures using the first algorithm suite, and ignore Signature 2 in each Signature-Segment. If the Two-Algorithms flag is set to 1 and the BGPSEC speaker supports both algorithm suites, then the BGPSEC speaker follows the instructions below to validate the signatures using the first algorithm suite. The BGPSEC speaker MAY then analogously validate the second set of signatures using Algorithm Suite 2. If the BGPSEC speaker chooses to validate both sets of signatures, it returns "Good" if either the first or the second set of signatures successfully validate.

- (Step I): Locate the public key needed to verify the signature (in the current Signature-Segment). To do this, consult the valid RPKI end-entity certificate data and look for an SKI that matches the value in the Subject Key Identifier 1 field of the Signature-Segment. If no such SKI value is found in the valid RPKI data then validation fails and returns "Not Good". Similarly, if the SKI exists but the AS Number associated with the SKI does NOT match the AS Number in the Signature-Segment, then validation fails and returns "Not Good".

- (Step II): Compute the digest function (for Algorithm Suite 1) on the appropriate data. If the segment is not the (least recently added) segment corresponding to the origin AS, then the digest function should be computed on the following sequence of octets:

Sequence of Octets to be Hashed

```
+-----------------------------------------------------------+
| Signature 1 Length Field in the Next Segment  (1 octet) |
+-----------------------------------------------------------+
| Signature 1 Field in the Next Segment         (variable) |
+-----------------------------------------------------------+
| pCount Field in the Current Segment           (1 octet)  |
+-----------------------------------------------------------+
| AS Number of Previous AS                      (4 octets) |
+-----------------------------------------------------------+
```

The ’Signature 1 Field in the Next Segment’ and ’Signature 1 Length Field in Next Segment’ are the Signature 1 field and Signature 1 Length fields found in the Signature-Segment that is next to be processed (that is, the next most recently added Signature- Segment). The ’pCount Field in the Current Segment’ is the pCount field found in the Signature-Segment that is currently being processed.

For the first segment to be processed (the most recently added segment), the ’AS Number of Subsequent AS’ is the AS number of the BGPSEC speaker validating the update message. Note that if a BGPSEC speaker uses multiple AS Numbers (e.g., the BGPSEC speaker is a member of a confederation), the AS number used here MUST be the AS number announced in the OPEN message for the BGP session over which the BGPSEC update was received.

For each other Signature-Segment, the ’AS Number of Previous AS’ is the AS number in the Signature-Segment that was most recently processed.

Alternatively, if the segment being processed corresponds to the origin AS, then the digest function should be computed on the following sequence of octets:

```
                Sequence of Octets to be Hashed
           ------------------------------------------+
           | AS Number of Previous AS    (4 octets)   |
           +------------------------------------------+
           | Origin AS Number            (4 octets)   |
           +------------------------------------------+
           | Algorithm Suite 1 Identifier  (1 octet)  |
           +------------------------------------------+
           | pCount      (1 octet)                    |
           +------------------------------------------+
           | NLRI Length  (1 octet)                   |
           +------------------------------------------+
           | NLRI Prefix  (variable)                  |
           +------------------------------------------+
```

The NLRI Length, NLRI Prefix, Expire Time, and Algorithm Suite Identifier are all obtained in a straight forward manner from the NLRI of the update message or the BGPSEC_Path_Signatures attribute being validated. The pCount field is taken from the Signature-Segment currently being processed.

The Origin AS Number is the same Origin AS Number that was located in Step I above. (That is, the AS number in the least recently added Signature-Segment.)

The ’AS Number of Previous AS’ is the AS Number in the Signature-Segment that was most recently processed (i.e., processed before the current segment).

- (Step III): Use the signature validation algorithm (for the given algorithm suite) to verify the signature in the current segment. That is, invoke the signature validation algorithm on the following three inputs: the value of the Signature field in the current segment; the digest value computed in Step II above; and the public key obtained from the valid RPKI data in Step I above. If the signature validation algorithm determines that the signature is invalid, validation has failed and return ’Not Good’. If the signature validation algorithm determines that the signature is valid, then continue processing Signature-Segments.

If all Signature-Segments pass validation (i.e., all segments are processed and the algorithm has not yet returned ’Not Good’), then validation succeeds and returns ’Good’.

6. Algorithms and Extensibility

6.1. Algorithm Suite Considerations

Note that there is currently no support for bilateral negotiation between BGPSEC peers to use of a particular (digest and signature) algorithm suite using BGP capabilities. This is because the algorithm suite used by the sender of a BGPSEC update message must be understood not only by the peer to whom he is directly sending the message, but also by all BGPSEC speakers to whom the route advertisement is eventually propagated. Therefore, selection of an algorithm suite cannot be a local matter negotiated by BGP peers, but instead must be coordinated throughout the Internet.

To this end, a mandatory algorithm suites document will be created which specifies a mandatory-to-use ’current’ algorithm suite for use by all BGPSEC speakers. Additionally, the document specifies an additional ’new’ algorithm suite that is recommended to implement.

It is anticipated that in the future the mandatory algorithm suites document will be updated to specify a transition from the ’current’ algorithm suite to the ’new’ algorithm suite. During the period of transition (likely a small number of years), all BGPSEC update messages SHOULD simultaneously use both the ’current’ algorithm suite and the ’new’ algorithm suite. (Note that Sections 3 and 4 specify how the BGPSEC_Path_Signatures attribute can contain signatures, in parallel, for two algorithm suites.) Once the transition is complete, use of the old ’current’ algorithm will be deprecated, use of the ’new’ algorithm will be mandatory, and a subsequent ’even newer’ algorithm suite may be specified as recommend to implement. Once the transition has successfully been completed in this manner, BGPSEC speakers SHOULD include only a signatures corresponding to the ’new’ algorithm.

6.2. Extensibility Considerations

This section discusses potential changes to BGPSEC that would require substantial changes to the processing of the BGPSEC_Path_Signatures and thus necessitate a new version of BGPSEC. Examples of such changes include

- A new type of signature algorithm for which the number of signatures in the Signature-List Block is not equal to the number of ASes in the AS_PATH (e.g., aggregate signatures)

- Changes to the data that is protected by the BGPSEC signatures (e.g., protection of attributes other than AS_PATH)

In the case that such a change to BGPSEC were deemed desirable, it is expected that a subsequent version of BGPSEC would be created and that this version of BGPSEC would specify a new BGP Path Attribute, let’s call it BGPSEC_PATH_SIG_TWO, which is designed to accommodate the desired changes to BGPSEC. In such a case, the mandatory algorithm suites document would be updated to specify algorithm suites appropriate for the new version of BGPSEC.

At this point a transition would begin which is analogous to the algorithm transition discussed in Section 6.2. During the transition period all BGPSEC speakers SHOULD simultaneously include both the BGPSEC_PATH_SIGNATURES attribute and the new BGPSEC_PATH_SIG_TWO attribute. Once the transition is complete, the use of BGPSEC_PATH_SIGNATURES could then be deprecated, at which point BGPSEC speakers SHOULD include only the new BGPSEC_PATH_SIG_TWO attribute. Such a process could facilitate a transition to a new BGPSEC semantics in a backwards compatible fashion.

7. Security Considerations

For discussion of the BGPSEC threat model and related security considerations, please see [8].

A BGPSEC speaker who receives a valid BGPSEC update message, containing a route advertisement for a given prefix, is provided with the following security guarantees:

- The origin AS number corresponds to an autonomous system that has been authorized by the IP address space holder to originate route advertisements for the given prefix.

- For each subsequent AS number in the AS-Path, a BGPSEC speaker authorized by the holder of the AS number selected the given route as the best route to the given prefix.

- For each AS number in the AS Path, a BGPSEC speaker authorized by the holder of the AS number intentionally propagated the route advertisement to the next AS in the AS-Path.

That is, the recipient of a valid BGPSEC Update message is assured that the AS-Path corresponds to a sequence of autonomous systems who have all agreed in principle to forward packets to the given prefix along the indicated path. (It should be noted BGPSEC does not offer a precise guarantee that the data packets would propagate along the indicated path; it only guarantees that the BGP update conveying the path indeed propagated along the indicated path.) Furthermore, the recipient is assured that this path terminates in an autonomous system that has been authorized by the IP address space holder as a legitimate destination for traffic to the given prefix.

Note that although BGPSEC provides a mechanism for an AS to validate that a received update message has certain security properties, the use of such a mechanism to influence route selection is completely a matter of local policy. Therefore, a BGPSEC speaker can make no assumptions about the validity of a route received from an external BGPSEC peer. That is, a compliant BGPSEC peer may (depending on the local policy of the peer) send update messages that fail the validity test in Section 5. Thus, a BGPSEC speaker MUST completely validate all BGPSEC update messages received from external peers. (Validation of update messages received from internal peers is a matter of local policy, see Section 5).

Note that there may be cases where a BGPSEC speaker deems 'Good' (as per the validation algorithm in Section 5.1) a BGPSEC update message that contains two sets of signatures, one 'Good' and one 'Not Good'. That is, the update message contains two sets of signatures corresponding to two algorithm suites, and one set of signatures verifies correctly and the other set of signatures fails to verify. In this case, the protocol specifies that if the BGPSEC speaker propagates the route advertisement received in such an update message then the BGPSEC speaker SHOULD add its signature using both the algorithm suites. Thus the BGPSEC speaker creates a signature using both algorithm suites and creates a new update message that contains both the 'Good' and the 'Not Good' set of signatures (from its own vantage point).

To understand the reason for such a design decision consider the case where the BGPSEC speaker receives an update message with both a set of algorithm A signatures which are 'Good' and a set of algorithm B signatures which are 'Not Good'. In such a case it is possible (perhaps even quite likely) that some of the BGPSEC speaker's peers (or other entities further 'downstream' in the BGP topology) do not support algorithm A. Therefore, if the BGPSEC speaker were to remove the 'Not Good' set of signatures corresponding to algorithm B, such entities would treat the message as though it were unsigned. By including the 'Not Good' set of signatures when propagating a route advertisement, the BGPSEC speaker ensures that 'downstream' entities have as much information as possible to make an informed opinion about the validation status of a BGPSEC update.

Note also that during a period of partial BGPSEC deployment, a 'downstream' entity might reasonably treat unsigned messages different from BGPSEC updates that contain a single set of 'Not Good' signatures. That is, by removing the set of 'Not Good' signatures the BGPSEC speaker might actually cause a downstream entity to 'upgrade' the status of a route advertisement from 'Not Good' to unsigned. Finally, note that in the above scenario, the BGPSEC speaker might have deemed algorithm A signatures 'Good' only because of some issue with RPKI state local to his AS (for example, his AS might not yet have obtained a CRL indicating that a key used to verify an algorithm A signature belongs to a newly revoked certificate). In such a case, it is highly desirable for a downstream entity to treat the update as 'Not Good' (due to the revocation) and not as 'unsigned' (which would happen if the 'Not Good' signatures were removed).

A similar argument applies to the case where a BGPSEC speaker (for some reason such as lack of viable alternatives) selects as his best route to a given prefix a route obtained via a 'Not Good' BGPSEC update message. (That is, a BGPSEC update containing only 'Not Good' signatures.) In such a case, the BGPSEC speaker should propagate a signed BGPSEC update message, adding his signature to the 'Not Good' signatures that already exist. Again, this is to ensure that 'downstream' entities are able to make an informed decision and not erroneously treat the route as unsigned. It may also be noted here that due to possible differences in RPKI data at different vantage points in the network, a BGPSEC update that was deemed 'Not Good' at an upstream BGPSEC speaker may indeed be deemed 'Good' at another BGP speaker downstream.

Therefore, it is important to note that when a BGPSEC speaker signs an outgoing update message, it is not attesting to a belief that all signatures prior to its are valid. Instead it is merely asserting that:

1. The BGPSEC speaker received the given route advertisement with the indicated NLRI and AS Path;

2. The BGPSEC speaker selected this route as the best route to the given prefix; and

3. The BGPSEC speaker chose to propagate an advertisement for this route to the peer (implicitly) indicated by the 'Target AS'

The BGPSEC update validation procedure is a potential target for denial of service attacks against a BGPSEC speaker. To mitigate the effectiveness of such denial of service attacks, BGPSEC speakers should implement an update validation algorithm that performs expensive checks (e.g., signature verification) after less expensive checks (e.g., syntax checks). The validation algorithm specified in Section 5.1 was chosen so as to perform checks which are likely to be expensive after checks that are likely to be inexpensive. However, the relative cost of performing required validation steps may vary between implementations, and thus the algorithm specified in Section 5.1 may not provide the best denial of service protection for all implementations.

Finally, the mechanism of setting the pCount field to zero is included in this specification to enable route servers in the control path to participate in BGPSEC without increasing the effective length of the AS_PATH. However, entities other than route servers could conceivably use this mechanism (set the pCount to zero) to attract traffic (by reducing the effective length of the AS_PATH) illegitimately. This risk is largely mitigated if every BGPSEC speaker drops incoming update messages that set pCount to zero but come from a peer that is not a route server. However, note that a recipient of a BGPSEC update message in which an upstream entity that is two or more hops away set pCount to zero is unable to verify for themselves whether pCount was set to zero legitimately.

8. Contributors

8.1. Authors

Rob Austein
Dragon Research Labs
sra@hactrn.net

Steven Bellovin
Columbia University
smb@cs.columbia.edu

Randy Bush
Internet Initiative Japan
randy@psg.com

Russ Housley
Vigil Security
housley@vigilsec.com

Matt Lepinski
BBN Technologies
lepinski@bbn.com

Stephen Kent
BBN Technologies
kent@bbn.com

Warren Kumari
Google
warren@kumari.net

Doug Montgomery
USA National Institute of Standards and Technology
dougm@nist.gov

Kotikalapudi Sriram
USA National Institute of Standards and Technology
kotikalapudi.sriram@nist.gov

Samuel Weiler
Cobham
weiler+ietf@watson.org

8.2. Acknowledgements

The authors would like to thank Luke Berndt, Wes George, Sharon Goldberg, Ed Kern, Chris Morrow, Doug Maughan, Pradosh Mohapatra, Russ Mundy, Sandy Murphy, Keyur Patel, Mark Reynolds, Heather Schiller, Jason Schiller, John Scudder, Ruediger Volk and David Ward for their valuable input and review.

9. Normative References

[1] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4", RFC 4271, January 2006.

[2] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

[3] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, February 2009.

[4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[5] Patel, K., Ward, D., and R. Bush, "Extended Message support for BGP", March 2011.

[6] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations", February 2011.

[7] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", February 2011.

[8] Kent, S., "Threat Model for BGP Path Security", June 2011.

[9] Bush, R., "BGPsec Operational Considerations", October 2011.

[10] Turner, S., "BGP Algorithms, Key Formats, & Signature Formats", December 2011.

[11] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", December 2011.

[12] Bush, R. and R. Austein, "The RPKI/Router Protocol", October 2011.

Author's Address

Matthew Lepinski (editor)
BBN
10 Moulton St
Cambridge, MA 55409
US

Phone: +1 617 873 5939
Email: mlepinski@bbn.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2017

M. Lepinski, Ed.
NCF
K. Sriram, Ed.
NIST

April 27, 2017

BGPsec Protocol Specification
draft-ietf-sidr-bgpsec-protocol-23

Abstract

This document describes BGPsec, an extension to the Border Gateway Protocol (BGP) that provides security for the path of autonomous systems (ASes) through which a BGP update message passes. BGPsec is implemented via an optional non-transitive BGP path attribute that carries digital signatures produced by each autonomous system that propagates the update message. The digital signatures provide confidence that every AS on the path of ASes listed in the update message has explicitly authorized the advertisement of the route.

Status of This Memo

This Internet-Draft will expire on October 29, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document describes BGPsec, a mechanism for providing path security for Border Gateway Protocol (BGP) [RFC4271] route advertisements. That is, a BGP speaker who receives a valid BGPsec update has cryptographic assurance that the advertised route has the following property: Every AS on the path of ASes listed in the update message has explicitly authorized the advertisement of the route to the subsequent AS in the path.

This document specifies an optional (non-transitive) BGP path attribute, BGPsec_Path. It also describes how a BGPsec-compliant BGP speaker (referred to hereafter as a BGPsec speaker) can generate, propagate, and validate BGP update messages containing this attribute to obtain the above assurances.

BGPsec is intended to be used to supplement BGP Origin Validation [RFC6483][RFC6811] and when used in conjunction with origin validation, it is possible to prevent a wide variety of route hijacking attacks against BGP.

BGPsec relies on the Resource Public Key Infrastructure (RPKI) certificates that attest to the allocation of AS number and IP address resources. (For more information on the RPKI, see RFC 6480 [RFC6480] and the documents referenced therein.) Any BGPsec speaker who wishes to send, to external (eBGP) peers, BGP update messages containing the BGPsec_Path needs to possess a private key associated with an RPKI router certificate [I-D.ietf-sidr-bgpsec-pki-profiles] that corresponds to the BGPsec speaker's AS number. Note, however, that a BGPsec speaker does not need such a certificate in order to validate received update messages containing the BGPsec_Path attribute (see Section 5.2).

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. BGPsec Negotiation

This document defines a BGP capability [RFC5492] that allows a BGP speaker to advertise to a neighbor the ability to send or to receive BGPsec update messages (i.e., update messages containing the BGPsec_Path attribute).

2.1. The BGPsec Capability

This capability has capability code: TBD

The capability length for this capability MUST be set to 3.

The three octets of the capability format are specified in Figure 1.

```
                   0   1   2   3      4      5   6   7
                 +---------------------------------------+
                 | Version          | Dir |  Unassigned  |
                 +---------------------------------------+
                 |                                       |
                 +------           AFI              -----+
                 |                                       |
                 +---------------------------------------+
```

Figure 1: BGPsec Capability format.

The first four bits of the first octet indicate the version of BGPsec for which the BGP speaker is advertising support. This document defines only BGPsec version 0 (all four bits set to zero). Other versions of BGPsec may be defined in future documents. A BGPsec speaker MAY advertise support for multiple versions of BGPsec by including multiple versions of the BGPsec capability in its BGP OPEN message.

The fifth bit of the first octet is a direction bit which indicates whether the BGP speaker is advertising the capability to send BGPsec update messages or receive BGPsec update messages. The BGP speaker sets this bit to 0 to indicate the capability to receive BGPsec update messages. The BGP speaker sets this bit to 1 to indicate the capability to send BGPsec update messages.

The remaining three bits of the first octet are unassigned and for future use. These bits are set to zero by the sender of the capability and ignored by the receiver of the capability.

The second and third octets contain the 16-bit Address Family Identifier (AFI) which indicates the address family for which the BGPsec speaker is advertising support for BGPsec. This document only specifies BGPsec for use with two address families, IPv4 and IPv6, AFI values 1 and 2 respectively [IANA-AF]. BGPsec for use with other address families may be specified in future documents.

2.2. Negotiating BGPsec Support

In order to indicate that a BGP speaker is willing to send BGPsec update messages (for a particular address family), a BGP speaker sends the BGPsec Capability (see Section 2.1) with the Direction bit (the fifth bit of the first octet) set to 1. In order to indicate that the speaker is willing to receive BGP update messages containing the BGPsec_Path attribute (for a particular address family), a BGP speaker sends the BGPsec capability with the Direction bit set to 0. In order to advertise the capability to both send and receive BGPsec update messages, the BGP speaker sends two copies of the BGPsec capability (one with the direction bit set to 0 and one with the direction bit set to 1).

Similarly, if a BGP speaker wishes to use BGPsec with two different address families (i.e., IPv4 and IPv6) over the same BGP session, then the speaker includes two instances of this capability (one for each address family) in the BGP OPEN message. A BGP speaker MUST NOT announce BGPsec capability if it does not support the BGP multiprotocol extension [RFC4760]. Additionally, a BGP speaker MUST NOT advertise the capability of BGPsec support for a particular AFI unless it has also advertised the multiprotocol extension capability for the same AFI [RFC4760].

In a BGPsec peering session, a peer is permitted to send update messages containing the BGPsec_Path attribute if, and only if:

- The given peer sent the BGPsec capability for a particular version of BGPsec and a particular address family with the Direction bit set to 1; and
- The other (receiving) peer sent the BGPsec capability for the same version of BGPsec and the same address family with the Direction bit set to 0.

In such a session, it can be said that the use of the particular version of BGPsec has been negotiated for a particular address family. Traditional BGP update messages (i.e. unsigned, containing AS_PATH attribute) MAY be sent within a session regardless of whether or not the use of BGPsec is successfully negotiated. However, if BGPsec is not successfully negotiated, then BGP update messages containing the BGPsec_Path attribute MUST NOT be sent.

This document defines the behavior of implementations in the case where BGPsec version zero is the only version that has been successfully negotiated. Any future document which specifies additional versions of BGPsec will need to specify behavior in the case that support for multiple versions is negotiated.

BGPsec cannot provide meaningful security guarantees without support for four-byte AS numbers. Therefore, any BGP speaker that announces the BGPsec capability, MUST also announce the capability for four-byte AS support [RFC6793]. If a BGP speaker sends the BGPsec capability but not the four-byte AS support capability then BGPsec has not been successfully negotiated, and update messages containing the BGPsec_Path attribute MUST NOT be sent within such a session.

3. The BGPsec_Path Attribute

The BGPsec_Path attribute is an optional non-transitive BGP path attribute.

This document registers an attribute type code for this attribute: BGPsec_Path (see Section 9).

The BGPsec_Path attribute carries the secured information regarding the path of ASes through which an update message passes. This includes the digital signatures used to protect the path information. The update messages that contain the BGPsec_Path attribute are referred to as "BGPsec Update messages". The BGPsec_Path attribute replaces the AS_PATH attribute in a BGPsec update message. That is, update messages that contain the BGPsec_Path attribute MUST NOT contain the AS_PATH attribute, and vice versa.

The BGPsec_Path attribute is made up of several parts. The high-level diagram in Figure 2 provides an overview of the structure of the BGPsec_Path attribute.

```
        +---------------------------------------------------------+
        |     +-----------------+                                 |
        |     |   Secure Path   |                                 |
        |     +-----------------+                                 |
        |     |    pCount X     |                                 |
        |     |    Flags X      |                                 |
        |     |    AS X         |                                 |
        |     |    pCount Y     |                                 |
        |     |    Flags Y      |                                 |
        |     |    AS Y         |                                 |
        |     |      ...        |                                 |
        |     +-----------------+                                 |
        |                                                         |
        |     +-----------------+       +-----------------+       |
        |     | Sig Block 1     |       |  Sig Block 2    |       |
        |     +-----------------+       +-----------------+       |
        |     | Alg Suite 1     |       |  Alg Suite 2    |       |
        |     | SKI X1          |       |  SKI X2         |       |
        |     | Signature X1    |       |  Signature X2   |       |
        |     | SKI Y1          |       |  SKI Y2         |       |
        |     | Signature Y1    |       |  Signature Y2   |       |
        |     |      ...        |       |      ....       |       |
        |     +-----------------+       +-----------------+       |
        |                                                         |
        +---------------------------------------------------------+
```

Figure 2: High-level diagram of the BGPsec_Path attribute.

Figure 3 provides the specification of the format for the BGPsec_Path attribute.

```
         +-------------------------------------------------------+
         | Secure_Path                             (variable)    |
         +-------------------------------------------------------+
         | Sequence of one or two Signature_Blocks (variable)    |
         +-------------------------------------------------------+
```

Figure 3: BGPsec_Path attribute format.

The Secure_Path contains AS path information for the BGPsec update message. This is logically equivalent to the information that is contained in a non-BGPsec AS_PATH attribute. The information in Secure_Path is used by BGPsec speakers in the same way that information from the AS_PATH is used by non-BGPsec speakers. The format of the Secure_Path is described below in Section 3.1.

The BGPsec_Path attribute will contain one or two Signature_Blocks, each of which corresponds to a different algorithm suite. Each of the Signature_Blocks will contain a Signature Segment for each AS number (i.e., Secure_Path Segment) in the Secure_Path. In the most common case, the BGPsec_Path attribute will contain only a single Signature_Block. However, in order to enable a transition from an old algorithm suite to a new algorithm suite (without a flag day), it will be necessary to include two Signature_Blocks (one for the old algorithm suite and one for the new algorithm suite) during the transition period. (See Section 6.1 for more discussion of algorithm transitions.) The format of the Signature_Blocks is described below in Section 3.2.

3.1. Secure_Path

A detailed description of the Secure_Path information in the BGPsec_Path attribute is provided here.

```
+-----------------------------------------------+
| Secure_Path Length                (2 octets) |
+-----------------------------------------------+
| One or More Secure_Path Segments   (variable) |
+-----------------------------------------------+
```

Figure 4: Secure_Path format.

The specification for the Secure_Path field is provided in Figure 4 and Figure 5. The Secure_Path Length contains the length (in octets) of the entire Secure_Path (including the two octets used to express this length field). As explained below, each Secure_Path Segment is six octets long. Note that this means the Secure_Path Length is two greater than six times the number Secure_Path Segments (i.e., the number of AS numbers in the path).

The Secure_Path contains one Secure_Path Segment (see Figure 5) for each Autonomous System in the path to the originating AS of the prefix specified in the update message. (Note: Repeated Autonomous Systems are compressed out using the pCount field as discussed below.)

```
+-----------------------------------------------------+
| pCount         (1 octet)                            |
+-----------------------------------------------------+
| Confed_Segment flag (1 bit) |  Unassigned (7 bits)   | (Flags)
+-----------------------------------------------------+
| AS Number      (4 octets)                           |
+-----------------------------------------------------+
```

Figure 5: Secure_Path Segment format.

The AS Number (in Figure 5) is the AS number of the BGP speaker that added this Secure_Path Segment to the BGPsec_Path attribute. (See Section 4 for more information on populating this field.)

The pCount field contains the number of repetitions of the associated autonomous system number that the signature covers. This field enables a BGPsec speaker to mimic the semantics of prepending multiple copies of their AS to the AS_PATH without requiring the speaker to generate multiple signatures. Note that Section 9.1.2.2 ("Breaking Ties") in [RFC4271] mentions "number of AS numbers" in the AS_PATH attribute that is used in the route selection process. This metric (number of AS numbers) is the same as the AS path length obtained in BGPsec by summing the pCount values in the BGPsec_Path attribute. The pCount field is also useful in managing route servers (see Section 4.2), AS confederations (see Section 4.3), and AS Number migrations (see [I-D.ietf-sidr-as-migration] for details).

The left most (i.e. the most significant) bit of the Flags field in Figure 5 is the Confed_Segment flag. The Confed_Segment flag is set to one to indicate that the BGPsec speaker that constructed this Secure_Path Segment is sending the update message to a peer AS within the same Autonomous System confederation [RFC5065]. (That is, a sequence of consecutive Confed_Segment flags are set in a BGPsec update message whenever, in a non-BGPsec update message, an AS_PATH segment of type AS_CONFED_SEQUENCE occurs.) In all other cases the Confed_Segment flag is set to zero.

The remaining seven bits of the Flags are unassigned and MUST be set to zero by the sender, and ignored by the receiver. Note, however, that the signature is computed over all eight bits of the flags field.

As stated earlier in Section 2.2, BGPsec peering requires that the peering ASes MUST each support four-byte AS numbers. Currently-assigned two-byte AS numbers are converted into four-byte AS numbers by setting the two high-order octets of the four-octet field to zero [RFC6793].

3.2. Signature_Block

A detailed description of the Signature_Blocks in the BGPsec_Path attribute is provided here using Figure 6 and Figure 7.

```
+---------------------------------------------+
| Signature_Block Length         (2 octets)   |
+---------------------------------------------+
| Algorithm Suite Identifier     (1 octet)    |
+---------------------------------------------+
| Sequence of Signature Segments (variable)   |
+---------------------------------------------+
```

Figure 6: Signature_Block format.

The Signature_Block Length in Figure 6 is the total number of octets in the Signature_Block (including the two octets used to express this length field).

The Algorithm Suite Identifier is a one-octet identifier specifying the digest algorithm and digital signature algorithm used to produce the digital signature in each Signature Segment. An IANA registry of algorithm identifiers for use in BGPsec is specified in the BGPsec algorithms document [I-D.ietf-sidr-bgpsec-algs].

A Signature_Block in Figure 6 has exactly one Signature Segment (see Figure 7) for each Secure_Path Segment in the Secure_Path portion of the BGPsec_Path Attribute. (That is, one Signature Segment for each distinct AS on the path for the prefix in the Update message.)

```
+---------------------------------------------+
| Subject Key Identifier (SKI)  (20 octets)   |
+---------------------------------------------+
| Signature Length              (2 octets)    |
+---------------------------------------------+
| Signature                     (variable)    |
+---------------------------------------------+
```

Figure 7: Signature Segment format.

The Subject Key Identifier (SKI) field in Figure 7 contains the value in the Subject Key Identifier extension of the RPKI router certificate [RFC6487] that is used to verify the signature (see Section 5 for details on validity of BGPsec update messages). The SKI field has a fixed 20 octets size. See Section 6.2 for considerations for the SKI size.

The Signature Length field contains the size (in octets) of the value in the Signature field of the Signature Segment.

The Signature in Figure 7 contains a digital signature that protects the prefix and the BGPsec_Path attribute (see Section 4 and Section 5 for details on signature generation and validation, respectively).

4. BGPsec Update Messages

Section 4.1 provides general guidance on the creation of BGPsec Update Messages -- that is, update messages containing the BGPsec_Path attribute.

Section 4.2 specifies how a BGPsec speaker generates the BGPsec_Path attribute to include in a BGPsec Update message.

Section 4.3 contains special processing instructions for members of an autonomous system confederation [RFC5065]. A BGPsec speaker that is not a member of such a confederation MUST NOT set the Confed_Segment flag in its Secure_Path Segment (i.e. leave the flag bit at default value zero) in all BGPsec update messages it sends.

Section 4.4 contains instructions for reconstructing the AS_PATH attribute in cases where a BGPsec speaker receives an update message with a BGPsec_Path attribute and wishes to propagate the update message to a peer who does not support BGPsec.

4.1. General Guidance

The information protected by the signature on a BGPsec update message includes the AS number of the peer to whom the update message is being sent. Therefore, if a BGPsec speaker wishes to send a BGPsec update to multiple BGP peers, it MUST generate a separate BGPsec update message for each unique peer AS to whom the update message is sent.

A BGPsec update message MUST advertise a route to only a single prefix. This is because a BGPsec speaker receiving an update message with multiple prefixes would be unable to construct a valid BGPsec update message (i.e., valid path signatures) containing a subset of the prefixes in the received update. If a BGPsec speaker wishes to advertise routes to multiple prefixes, then it MUST generate a separate BGPsec update message for each prefix. Additionally, a BGPsec update message MUST use the MP_REACH_NLRI [RFC4760] attribute to encode the prefix.

The BGPsec_Path attribute and the AS_PATH attribute are mutually exclusive. That is, any update message containing the BGPsec_Path attribute MUST NOT contain the AS_PATH attribute. The information that would be contained in the AS_PATH attribute is instead conveyed in the Secure_Path portion of the BGPsec_Path attribute.

In order to create or add a new signature to a BGPsec update message with a given algorithm suite, the BGPsec speaker MUST possess a private key suitable for generating signatures for this algorithm suite. Additionally, this private key must correspond to the public key in a valid Resource PKI end-entity certificate whose AS number resource extension includes the BGPsec speaker’s AS number [I-D.ietf-sidr-bgpsec-pki-profiles]. Note also that new signatures are only added to a BGPsec update message when a BGPsec speaker is generating an update message to send to an external peer (i.e., when the AS number of the peer is not equal to the BGPsec speaker’s own AS number).

The Resource PKI enables the legitimate holder of IP address prefix(es) to issue a signed object, called a Route Origination Authorization (ROA), that authorizes a given AS to originate routes to a given set of prefixes (see RFC 6482 [RFC6482]). It is expected that most relying parties will utilize BGPsec in tandem with origin validation (see RFC 6483 [RFC6483] and RFC 6811 [RFC6811]). Therefore, it is RECOMMENDED that a BGPsec speaker only originate a BGPsec update advertising a route for a given prefix if there exists a valid ROA authorizing the BGPsec speaker’s AS to originate routes to this prefix.

If a BGPsec router has received only a non-BGPsec update message containing the AS_PATH attribute (instead of the BGPsec_Path attribute) from a peer for a given prefix, then it MUST NOT attach a BGPsec_Path attribute when it propagates the update message. (Note that a BGPsec router may also receive a non-BGPsec update message from an internal peer without the AS_PATH attribute, i.e., with just the NLRI in it. In that case, the prefix is originating from that AS, and if it is selected for advertisement, the BGPsec speaker SHOULD attach a BGPsec_Path attribute and send a signed route (for that prefix) to its external BGPsec-speaking peers.)

Conversely, if a BGPsec router has received a BGPsec update message (with the BGPsec_Path attribute) from a peer for a given prefix and it chooses to propagate that peer’s route for the prefix, then it SHOULD propagate the route as a BGPsec update message containing the BGPsec_Path attribute.

Note that removing BGPsec signatures (i.e., propagating a route advertisement without the BGPsec_Path attribute) has significant security ramifications. (See Section 8 for discussion of the security ramifications of removing BGPsec signatures.) Therefore, when a route advertisement is received via a BGPsec update message, propagating the route advertisement without the BGPsec_Path attribute is NOT RECOMMENDED, unless the message is sent to a peer that did not advertise the capability to receive BGPsec update messages (see Section 4.4).

Furthermore, note that when a BGPsec speaker propagates a route advertisement with the BGPsec_Path attribute it is not attesting to the validation state of the update message it received. (See Section 8 for more discussion of the security semantics of BGPsec signatures.)

If the BGPsec speaker is producing an update message which would, in the absence of BGPsec, contain an AS_SET (e.g., the BGPsec speaker is performing proxy aggregation), then the BGPsec speaker MUST NOT include the BGPsec_Path attribute. In such a case, the BGPsec speaker MUST remove any existing BGPsec_Path in the received advertisement(s) for this prefix and produce a traditional (non-BGPsec) update message. It should be noted that BCP 172 [RFC6472] recommends against the use of AS_SET and AS_CONFED_SET in the AS_PATH of BGP updates.

The case where the BGPsec speaker sends a BGPsec update message to an iBGP peer is quite simple. When originating a new route advertisement and sending it to a BGPsec-capable iBGP peer, the BGPsec speaker omits the BGPsec_Path attribute. When originating a new route advertisement and sending it to a non-BGPsec iBGP peer, the BGPsec speaker includes an empty AS_PATH attribute in the update message. (An empty AS_PATH attribute is one whose length field contains the value zero [RFC4271].) When a BGPsec speaker chooses to forward a BGPsec update message to an iBGP peer, the BGPsec_Path attribute SHOULD NOT be removed, unless the peer doesn’t support BGPsec. In the case when an iBGP peer doesn’t support BGPsec, then a BGP update with AS_PATH is reconstructed from the BGPsec update and then forwarded (see Section 4.4). In particular, when forwarding to a BGPsec-capable iBGP (or eBGP) peer, the BGPsec_Path attribute SHOULD NOT be removed even in the case where the BGPsec update message has not been successfully validated. (See Section 5 for more information on validation, and Section 8 for the security ramifications of removing BGPsec signatures.)

All BGPsec update messages MUST conform to BGP’s maximum message size. If the resulting message exceeds the maximum message size, then the guidelines in Section 9.2 of RFC 4271 [RFC4271] MUST be followed.

4.2. Constructing the BGPsec_Path Attribute

When a BGPsec speaker receives a BGPsec update message containing a BGPsec_Path attribute (with one or more signatures) from an (internal or external) peer, it may choose to propagate the route advertisement by sending it to its other (internal or external) peers. When sending the route advertisement to an internal BGPsec-speaking peer, the BGPsec_Path attribute SHALL NOT be modified. When sending the route advertisement to an external BGPsec-speaking peer, the following procedures are used to form or update the BGPsec_Path attribute.

To generate the BGPsec_Path attribute on the outgoing update message, the BGPsec speaker first generates a new Secure_Path Segment. Note that if the BGPsec speaker is not the origin AS and there is an existing BGPsec_Path attribute, then the BGPsec speaker prepends its new Secure_Path Segment (places in first position) onto the existing Secure_Path.

The AS number in this Secure_Path Segment MUST match the AS number in the Subject field of the Resource PKI router certificate that will be used to verify the digital signature constructed by this BGPsec speaker (see Section 3.1.1 in [I-D.ietf-sidr-bgpsec-pki-profiles] and RFC 6487 [RFC6487]).

The pCount field of the Secure_Path Segment is typically set to the value 1. However, a BGPsec speaker may set the pCount field to a value greater than 1. Setting the pCount field to a value greater than one has the same semantics as repeating an AS number multiple times in the AS_PATH of a non-BGPsec update message (e.g., for traffic engineering purposes).

To prevent unnecessary processing load in the validation of BGPsec signatures, a BGPsec speaker SHOULD NOT produce multiple consecutive Secure_Path Segments with the same AS number. This means that to achieve the semantics of prepending the same AS number k times, a BGPsec speaker SHOULD produce a single Secure_Path Segment -- with pCount of k -- and a single corresponding Signature Segment.

A route server that participates in the BGP control plane, but does not act as a transit AS in the data plane, may choose to set pCount to 0. This option enables the route server to participate in BGPsec and obtain the associated security guarantees without increasing the length of the AS path. (Note that BGPsec speakers compute the length of the AS path by summing the pCount values in the BGPsec_Path attribute, see Section 5.) However, when a route server sets the pCount value to 0, it still inserts its AS number into the Secure_Path Segment, as this information is needed to validate the signature added by the route server. See [I-D.ietf-sidr-as-migration] for a discussion of setting pCount to 0 to facilitate AS Number Migration. Also, see Section 4.3 for the use of pCount=0 in the context of an AS confederation. See Section 7.2 for operational guidance for configuring a BGPsec router for setting pCount=0 and/or accepting pCount=0 from a peer.

Next, the BGPsec speaker generates one or two Signature_Blocks. Typically, a BGPsec speaker will use only a single algorithm suite, and thus create only a single Signature_Block in the BGPsec_Path attribute. However, to ensure backwards compatibility during a period of transition from a ’current’ algorithm suite to a ’new’ algorithm suite, it will be necessary to originate update messages that contain a Signature_Block for both the ’current’ and the ’new’ algorithm suites (see Section 6.1).

If the received BGPsec update message contains two Signature_Blocks and the BGPsec speaker supports both of the corresponding algorithm suites, then the new update message generated by the BGPsec speaker MUST include both of the Signature_Blocks. If the received BGPsec update message contains two Signature_Blocks and the BGPsec speaker only supports one of the two corresponding algorithm suites, then the BGPsec speaker MUST remove the Signature_Block corresponding to the algorithm suite that it does not understand. If the BGPsec speaker does not support the algorithm suites in any of the Signature_Blocks contained in the received update message, then the BGPsec speaker MUST NOT propagate the route advertisement with the BGPsec_Path attribute. (That is, if it chooses to propagate this route advertisement at all, it MUST do so as an unsigned BGP update message. See Section 4.4 for more information on converting to an unsigned BGP message.)

Note that in the case where the BGPsec_Path has two Signature_Blocks (corresponding to different algorithm suites), the validation algorithm (see Section 5.2) deems a BGPsec update message to be ’Valid’ if there is at least one supported algorithm suite (and corresponding Signature_Block) that is deemed ’Valid’. This means that a ’Valid’ BGPsec update message may contain a Signature_Block which is not deemed ’Valid’ (e.g., contains signatures that BGPsec does not successfully verify). Nonetheless, such Signature_Blocks MUST NOT be removed. (See Section 8 for a discussion of the security ramifications of this design choice.)

For each Signature_Block corresponding to an algorithm suite that the BGPsec speaker does support, the BGPsec speaker MUST add a new Signature Segment to the Signature_Block. This Signature Segment is prepended to the list of Signature Segments (placed in the first position) so that the list of Signature Segments appears in the same order as the corresponding Secure_Path Segments. The BGPsec speaker populates the fields of this new Signature Segment as follows.

The Subject Key Identifier field in the new segment is populated with the identifier contained in the Subject Key Identifier extension of the RPKI router certificate corresponding to the BGPsec speaker [I-D.ietf-sidr-bgpsec-pki-profiles]. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Signature field in the new segment contains a digital signature that binds the prefix and BGPsec_Path attribute to the RPKI router certificate corresponding to the BGPsec speaker. The digital signature is computed as follows:

- For clarity, let us number the Secure_Path and corresponding Signature Segments from 1 to N as follows. Let Secure_Path Segment 1 and Signature Segment 1 be the segments produced by the origin AS. Let Secure_Path Segment 2 and Signature Segment 2 be the segments added by the next AS after the origin. Continue this method of numbering and ultimately let Secure_Path Segment N and Signature Segment N be those that are being added by the current AS. The current AS (Nth AS) is signing and forwarding the update to the next AS (i.e. (N+1)th AS) in the chain of ASes that form the AS path.

- In order to construct the digital signature for Signature Segment N (the Signature Segment being produced by the current AS), first construct the sequence of octets to be hashed as shown in Figure 8. This sequence of octets includes all the data that the Nth AS attests to by adding its digital signature in the update which is being forwarded to a BGPsec speaker in the (N+1)th AS. (For the design rationale for choosing the specific structure in Figure 8, please see [Borchert].)

```
         +------------------------------------+
         | Target AS Number                   |
         +------------------------------------+ ---\
         | Signature Segment   : N-1          |     \
         +------------------------------------+     |
         | Secure_Path Segment : N            |     |
         +------------------------------------+     \
                ...                                  >  Data from
         +------------------------------------+     /   N Segments
         | Signature Segment   : 1            |     |
         +------------------------------------+     |
         | Secure_Path Segment : 2            |     |
         +------------------------------------+     /
         | Secure_Path Segment : 1            |    /
         +------------------------------------+---/
         | Algorithm Suite Identifier         |
         +------------------------------------+
         | AFI                                |
         +------------------------------------+
         | SAFI                               |
         +------------------------------------+
         | Prefix                             |
         +------------------------------------+
```

Figure 8: Sequence of octets to be hashed.

The elements in this sequence (Figure 8) MUST be ordered exactly as shown. The ’Target AS Number’ is the AS to whom the BGPsec speaker intends to send the update message. (Note that the ’Target AS Number’ is the AS number announced by the peer in the OPEN message of the BGP session within which the update is sent.) The Secure_Path and Signature Segments (1 through N-1) are obtained from the BGPsec_Path attribute. Finally, the Address Family Identifier (AFI), Subsequent Address Family Identifier (SAFI), and Prefix fields are obtained from the MP_REACH_NLRI attribute [RFC4760]. Additionally, in the Prefix field all of the trailing bits MUST be set to zero when constructing this sequence.

- Apply to this octet sequence (in Figure 8) the digest algorithm (for the algorithm suite of this Signature_Block) to obtain a digest value.

- Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature_Block) to obtain the digital signature. Then populate the Signature Field (in Figure 7) with this digital signature.

The Signature Length field (in Figure 7) is populated with the length (in octets) of the value in the Signature field.

4.3. Processing Instructions for Confederation Members

Members of autonomous system confederations [RFC5065] MUST additionally follow the instructions in this section for processing BGPsec update messages.

When a BGPsec speaker in an AS confederation receives a BGPsec update from a peer that is external to the confederation and chooses to propagate the update within the confederation, then it first adds a signature signed to its own Member-AS (i.e. the Target AS number is the BGPsec speaker’s Member-AS number). In this internally modified update, the newly added Secure_Path Segment contains the public AS number (i.e. Confederation Identifier), the Segment’s pCount value is set to 0, and Confed_Segment flag is set to one. Setting pCount=0 in this case helps ensure that the AS path length is not unnecessarily incremented. The newly added signature is generated using a private key corresponding to the public AS number of the confederation. The BGPsec speaker propagates the modified update to its peers within the confederation.

Any BGPsec_Path modifications mentioned below in the context of propagation of the update within the confederation are in addition to the modification described above (i.e. with pCount=0).

When a BGPsec speaker sends a BGPsec update message to a peer that belongs within its own Member-AS, the confederation member SHALL NOT modify the BGPsec_Path attribute. When a BGPsec speaker sends a BGPsec update message to a peer that is within the same confederation but in a different Member-AS, the BGPsec speaker puts its Member-AS number in the AS Number field of the Secure_Path Segment that it adds to the BGPsec update message. Additionally, in this case, the Member-AS that generates the Secure_Path Segment sets the Confed_Segment flag to one. Further, the signature is generated with a private key corresponding to the BGPsec speaker’s Member-AS Number. (Note: In this document, intra-Member-AS peering is regarded as iBGP and inter-Member-AS peering is regarded as eBGP. The latter is also known as confederation-eBGP.)

Within a confederation, the verification of BGPsec signatures added by other members of the confederation is optional. Note that if a confederation chooses not to verify digital signatures within the confederation, then BGPsec is able to provide no assurances about the integrity of the Member-AS Numbers placed in Secure_Path Segments where the Confed_Segment flag is set to one.

When a confederation member receives a BGPsec update message from a peer within the confederation and propagates it to a peer outside the confederation, it needs to remove all of the Secure_Path Segments added by confederation members as well as the corresponding Signature Segments. To do this, the confederation member propagating the route outside the confederation does the following:

- First, starting with the most recently added Secure_Path Segment, remove all of the consecutive Secure_Path Segments that have the Confed_Segment flag set to one. Stop this process once a Secure_Path Segment is reached which has its Confed_Segment flag set to zero. Keep a count of the number of segments removed in this fashion.

- Second, starting with the most recently added Signature Segment, remove a number of Signature Segments equal to the number of Secure_Path Segments removed in the previous step. (That is, remove the K most recently added Signature Segments, where K is the number of Secure_Path Segments removed in the previous step.)

- Finally, add a Secure_Path Segment containing, in the AS field, the AS Confederation Identifier (the public AS number of the confederation) as well as a corresponding Signature Segment. Note that all fields other than the AS field are populated as per Section 4.2.

Finally, as discussed above, an AS confederation MAY optionally decide that its members will not verify digital signatures added by members. In such a confederation, when a BGPsec speaker runs the algorithm in Section 5.2, the BGPsec speaker, during the process of Signature verifications, first checks whether the Confed_Segment flag in a Secure_Path Segment is set to one. If the flag is set to one, the BGPsec speaker skips the verification for the corresponding Signature, and immediately moves on to the next Secure_Path Segment. Note that as specified in Section 5.2, it is an error when a BGPsec speaker receives from a peer, who is not in the same AS confederation, a BGPsec update containing a Confed_Segment flag set to one.

4.4. Reconstructing the AS_PATH Attribute

BGPsec update messages do not contain the AS_PATH attribute. However, the AS_PATH attribute can be reconstructed from the BGPsec_Path attribute. This is necessary in the case where a route advertisement is received via a BGPsec update message and then propagated to a peer via a non-BGPsec update message (e.g., because the latter peer does not support BGPsec). Note that there may be additional cases where an implementation finds it useful to perform this reconstruction. Before attempting to reconstruct an AS_PATH for the purpose of forwarding an unsigned (non-BGPsec) update to a peer, a BGPsec speaker MUST perform the basic integrity checks listed in Section 5.2 to ensure that the received BGPsec update is properly formed.

The AS_PATH attribute can be constructed from the BGPsec_Path attribute as follows. Starting with a blank AS_PATH attribute, process the Secure_Path Segments in order from least-recently added (corresponding to the origin) to most-recently added. For each Secure_Path Segment perform the following steps:

1. If the Secure_Path Segment has pCount=0, then do nothing (i.e. move on to process the next Secure_Path Segment).

2. If the Secure_Path Segment has pCount greater than 0 and the Confed_Segment flag is set to one, then look at the most-recently added segment in the AS_PATH.

    * In the case where the AS_PATH is blank or in the case where the most-recently added segment is of type AS_SEQUENCE, add (prepend to the AS_PATH) a new AS_PATH segment of type AS_CONFED_SEQUENCE. This segment of type AS_CONFED_SEQUENCE shall contain a number of elements equal to the pCount field in the current Secure_Path Segment. Each of these elements shall be the AS number contained in the current Secure_Path Segment. (That is, if the pCount field is X, then the segment of type AS_CONFED_SEQUENCE contains X copies of the Secure_Path Segment’s AS Number field.)

    * In the case where the most-recently added segment in the AS_PATH is of type AS_CONFED_SEQUENCE then add (prepend to the segment) a number of elements equal to the pCount field in the current Secure_Path Segment. The value of each of these elements shall be the AS number contained in the current Secure_Path Segment. (That is, if the pCount field is X, then add X copies of the Secure_Path Segment’s AS Number field to the existing AS_CONFED_SEQUENCE.)

3. If the Secure_Path Segment has pCount greater than 0 and the Confed_Segment flag is set to zero, then look at the most-recently added segment in the AS_PATH.

    * In the case where the AS_PATH is blank or in the case where the most-recently added segment is of type AS_CONFED_SEQUENCE, add (prepend to the AS_PATH) a new AS_PATH segment of type AS_SEQUENCE. This segment of type AS_SEQUENCE shall contain a number of elements equal to the pCount field in the current

Secure_Path Segment. Each of these elements shall be the AS number contained in the current Secure_Path Segment. (That is, if the pCount field is X, then the segment of type AS_SEQUENCE contains X copies of the Secure_Path Segment's AS Number field.)

* In the case where the most recently added segment in the AS_PATH is of type AS_SEQUENCE then add (prepend to the segment) a number of elements equal to the pCount field in the current Secure_Path Segment. The value of each of these elements shall be the AS number contained in the current Secure_Path Segment. (That is, if the pCount field is X, then add X copies of the Secure_Path Segment's AS Number field to the existing AS_SEQUENCE.)

As part of the above described procedure, the following additional actions are performed in order not to exceed the size limitations of AS_SEQUENCE and AS_CONFED_SEQUENCE. While adding the next Secure_Path Segment (with its prepends, if any) to the AS_PATH being assembled, if it would cause the AS_SEQUENCE (or AS_CONFED_SEQUENCE) at hand to exceed the limit of 255 AS numbers per segment [RFC4271] [RFC5065], then the BGPsec speaker would follow the recommendations in RFC 4271 [RFC4271] and RFC 5065 [RFC5065] of creating another segment of the same type (AS_SEQUENCE or AS_CONFED_SEQUENCE) and continue filling that.

Finally, one special case of reconstruction of AS_PATH is when the BGPsec_Path attribute is absent. As explained in Section 4.1, when a BGPsec speaker originates a prefix and sends it to a BGPsec-capable iBGP peer, the BGPsec_Path is not attached. So when received from a BGPsec-capable iBGP peer, no BGPsec_Path attribute in a BGPsec update is equivalent to an empty AS_PATH [RFC4271].

5. Processing a Received BGPsec Update

Upon receiving a BGPsec update message from an external (eBGP) peer, a BGPsec speaker SHOULD validate the message to determine the authenticity of the path information contained in the BGPsec_Path attribute. Typically, a BGPsec speaker will also wish to perform origin validation (see RFC 6483 [RFC6483] and RFC 6811 [RFC6811]) on an incoming BGPsec update message, but such validation is independent of the validation described in this section.

Section 5.1 provides an overview of BGPsec validation and Section 5.2 provides a specific algorithm for performing such validation. (Note that an implementation need not follow the specific algorithm in Section 5.2 as long as the input/output behavior of the validation is identical to that of the algorithm in Section 5.2.) During exceptional conditions (e.g., the BGPsec speaker receives an incredibly large number of update messages at once) a BGPsec speaker MAY temporarily defer validation of incoming BGPsec update messages. The treatment of such BGPsec update messages, whose validation has been deferred, is a matter of local policy. However, an implementation SHOULD ensure that deferment of validation and status of deferred messages is visible to the operator.

The validity of BGPsec update messages is a function of the current RPKI state. When a BGPsec speaker learns that RPKI state has changed (e.g., from an RPKI validating cache via the RPKI-to-Router protocol [I-D.ietf-sidr-rpki-rtr-rfc6810-bis]), the BGPsec speaker MUST re-run validation on all affected update messages stored in its Adj-RIB-In [RFC4271]. For example, when a given RPKI router certificate ceases to be valid (e.g., it expires or is revoked), all update messages containing a signature whose SKI matches the SKI in the given certificate MUST be re-assessed to determine if they are still valid. If this reassessment determines that the validity state of an update has changed then, depending on local policy, it may be necessary to re-run best path selection.

BGPsec update messages do not contain an AS_PATH attribute. The Secure_Path contains AS path information for the BGPsec update message. Therefore, a BGPsec speaker MUST utilize the AS path information in the Secure_Path in all cases where it would otherwise use the AS path information in the AS_PATH attribute. The only exception to this rule is when AS path information must be updated in order to propagate a route to a peer (in which case the BGPsec speaker follows the instructions in Section 4). Section 4.4 provides an algorithm for constructing an AS_PATH attribute from a BGPsec_Path attribute. Whenever the use of AS path information is called for (e.g., loop detection, or use of AS path length in best path selection) the externally visible behavior of the implementation shall be the same as if the implementation had run the algorithm in Section 4.4 and used the resulting AS_PATH attribute as it would for a non-BGPsec update message.

5.1. Overview of BGPsec Validation

Validation of a BGPsec update message makes use of data from RPKI router certificates. In particular, it is necessary that the recipient have access to the following data obtained from valid RPKI router certificates: the AS Number, Public Key and Subject Key Identifier from each valid RPKI router certificate.

Note that the BGPsec speaker could perform the validation of RPKI router certificates on its own and extract the required data, or it could receive the same data from a trusted cache that performs RPKI validation on behalf of (some set of) BGPsec speakers. (For example, the trusted cache could deliver the necessary validity information to the BGPsec speaker using the router key PDU for the RPKI-to-Router protocol [I-D.ietf-sidr-rpki-rtr-rfc6810-bis].)

To validate a BGPsec update message containing the BGPsec_Path attribute, the recipient performs the validation steps specified in Section 5.2. The validation procedure results in one of two states: 'Valid' and 'Not Valid'.

It is expected that the output of the validation procedure will be used as an input to BGP route selection. That said, BGP route selection, and thus the handling of the validation states is a matter of local policy, and is handled using local policy mechanisms. Implementations SHOULD enable operators to set such local policy on a per-session basis. (That is, it is expected that some operators will choose to treat BGPsec validation status differently for update messages received over different BGP sessions.)

BGPsec validation needs only be performed at the eBGP edge. The validation status of a BGP signed/unsigned update MAY be conveyed via iBGP from an ingress edge router to an egress edge router via some mechanism, according to local policy within an AS. As discussed in Section 4, when a BGPsec speaker chooses to forward a (syntactically correct) BGPsec update message, it SHOULD be forwarded with its BGPsec_Path attribute intact (regardless of the validation state of the update message). Based entirely on local policy, an egress router receiving a BGPsec update message from within its own AS MAY choose to perform its own validation.

5.2. Validation Algorithm

This section specifies an algorithm for validation of BGPsec update messages. A conformant implementation MUST include a BGPsec update validation algorithm that is functionally equivalent to the externally visible behavior of this algorithm.

First, the recipient of a BGPsec update message performs a check to ensure that the message is properly formed. Both syntactical and protocol violation errors are checked. BGPsec_Path attribute MUST be present when a BGPsec update is received from an external (eBGP) BGPsec peer and also when such an update is propagated to an internal (iBGP) BGPsec peer (see Section 4.2). The error checks specified in Section 6.3 of [RFC4271] are performed, except that for BGPsec updates the checks on the AS_PATH attribute do not apply and instead the following checks on BGPsec_Path attribute are performed:

1. Check to ensure that the entire BGPsec_Path attribute is syntactically correct (conforms to the specification in this document).

2. Check that AS number in the most recently added Secure_Path Segment (i.e. the one corresponding to the eBGP peer from which the update message was received) matches the AS number of that peer as specified in the BGP OPEN message. (Note: This check is performed only at an ingress BGPsec routers where the update is first received from a peer AS.)

3. Check that each Signature_Block contains one Signature Segment for each Secure_Path Segment in the Secure_Path portion of the BGPsec_Path attribute. (Note that the entirety of each Signature_Block MUST be checked to ensure that it is well formed, even though the validation process may terminate before all signatures are cryptographically verified.)

4. Check that the update message does not contain an AS_PATH attribute.

5. If the update message was received from an BGPsec peer that is not a member of the BGPsec speaker’s AS confederation, check to ensure that none of the Secure_Path Segments contain a Flags field with the Confed_Segment flag set to one.

6. If the update message was received from a BGPsec peer that is a member of the BGPsec speaker’s AS confederation, check to ensure that the Secure_Path Segment corresponding to that peer contains a Flags field with the Confed_Segment flag set to one.

7. If the update message was received from a peer that is not expected to set pCount=0 (see Section 4.2 and Section 4.3) then check to ensure that the pCount field in the most-recently added Secure_Path Segment is not equal to zero. (Note: See router configuration guidance related to this in Section 7.2.)

8. Using the equivalent of AS_PATH corresponding to the Secure_Path in the update (see Section 4.4), check that the local AS number is not present in the AS path (i.e. rule out AS loop).

If any of these checks fail, it is an error in the BGPsec_Path attribute. BGPsec speakers MUST handle any syntactical or protocol errors in the BGPsec_Path attribute using the "treat-as-withdraw" approach as defined in RFC 7606 [RFC7606]. (Note: Since the AS number of a transparent route server does appear in the Secure_Path with pCount=0, the route server MAY check if its local AS is listed in the Secure_Path, and this check MAY be included in the loop detection check listed above.)

Next, the BGPsec speaker examines the Signature_Blocks in the BGPsec_Path attribute. A Signature_Block corresponding to an algorithm suite that the BGPsec speaker does not support is not considered in validation. If there is no Signature_Block corresponding to an algorithm suite that the BGPsec speaker supports, then in order to consider the update in the route selection process, the BGPsec speaker MUST strip the Signature_Block(s), reconstruct the AS_PATH from the Secure_Path (see Section 4.4), and treat the update as if it was received as an unsigned BGP update.

For each remaining Signature_Block (corresponding to an algorithm suite supported by the BGPsec speaker), the BGPsec speaker iterates through the Signature Segments in the Signature_Block, starting with the most recently added segment (and concluding with the least recently added segment). Note that there is a one-to-one correspondence between Signature Segments and Secure_Path Segments within the BGPsec_Path attribute. The following steps make use of this correspondence.

- (Step 1): Let there be K AS hops in a received BGPsec_Path attribute that is to be validated. Let AS(1), AS(2), ..., AS(K+1) denote the sequence of AS numbers from the origin AS to the validating AS. Let Secure_Path Segment N and Signature Segment N in the BGPsec_Path attribute refer to those corresponding to AS(N) (where N = 1, 2, ..., K). The BGPsec speaker that is processing and validating the BGPsec_Path attribute resides in AS(K+1). Let Signature Segment N be the Signature Segment that is currently being verified.

- (Step 2): Locate the public key needed to verify the signature (in the current Signature Segment). To do this, consult the valid RPKI router certificate data and look up all valid (AS, SKI, Public Key) triples in which the AS matches the AS number in the corresponding Secure_Path Segment. Of these triples that match the AS number, check whether there is an SKI that matches the value in the Subject Key Identifier field of the Signature Segment. If this check finds no such matching SKI value, then mark the entire Signature_Block as ’Not Valid’ and proceed to the next Signature_Block.

- (Step 3): Compute the digest function (for the given algorithm suite) on the appropriate data.

  In order to verify the digital signature in Signature Segment N, construct the sequence of octets to be hashed as shown in Figure 9 (using the notations defined in Step 1). (Note that this sequence is the same sequence that was used by AS(N) that created the Signature Segment N (see Section 4.2 and Figure 8).)

```
+------------------------------------+
| Target AS Number                   |
+------------------------------------+ ---\
| Signature Segment   : N-1          |     \
+------------------------------------+     |
| Secure_Path Segment : N            |     |
+------------------------------------+     \
       ...                                  >  Data from
+------------------------------------+     /   N Segments
| Signature Segment   : 1            |     |
+------------------------------------+     |
| Secure_Path Segment : 2            |     |
+------------------------------------+     /
| Secure_Path Segment : 1            |    /
+------------------------------------+---/
| Algorithm Suite Identifier         |
+------------------------------------+
| AFI                                |
+------------------------------------+
| SAFI                               |
+------------------------------------+
| Prefix                             |
+------------------------------------+
```

Figure 9: The Sequence of octets to be hashed for signature verification of Signature Segment N; N = 1,2, ..., K, where K is the number of AS hops in the BGPsec_Path attribute.

The elements in this sequence (Figure 9) MUST be ordered exactly as shown. For the first segment to be processed (the most recently added segment (i.e. N = K) given that there are K hops in the Secure_Path), the ’Target AS Number’ is AS(K+1), the AS number of the BGPsec speaker validating the update message. Note that if a BGPsec speaker uses multiple AS Numbers (e.g., the BGPsec speaker is a member of a confederation), the AS number used here MUST be the AS number announced in the OPEN message for the BGP session over which the BGPsec update was received.

For each other Signature Segment (N smaller than K), the ’Target AS Number’ is AS(N+1), the AS number in the Secure_Path Segment that corresponds to the Signature Segment added immediately after the one being processed. (That is, in the Secure_Path Segment that corresponds to the Signature Segment that the validator just finished processing.)

The Secure_Path and Signature Segment are obtained from the BGPsec_Path attribute. The Address Family Identifier (AFI), Subsequent Address Family Identifier (SAFI), and Prefix fields are obtained from the MP_REACH_NLRI attribute [RFC4760]. Additionally, in the Prefix field all of the trailing bits MUST be set to zero when constructing this sequence.

- (Step 4): Use the signature validation algorithm (for the given algorithm suite) to verify the signature in the current segment. That is, invoke the signature validation algorithm on the following three inputs: the value of the Signature field in the current segment; the digest value computed in Step 3 above; and the public key obtained from the valid RPKI data in Step 2 above. If the signature validation algorithm determines that the signature is invalid, then mark the entire Signature_Block as 'Not Valid' and proceed to the next Signature_Block. If the signature validation algorithm determines that the signature is valid, then continue processing Signature Segments (within the current Signature_Block).

If all Signature Segments within a Signature_Block pass validation (i.e., all segments are processed and the Signature_Block has not yet been marked 'Not Valid'), then the Signature_Block is marked as 'Valid'.

If at least one Signature_Block is marked as 'Valid', then the validation algorithm terminates and the BGPsec update message is deemed to be 'Valid'. (That is, if a BGPsec update message contains two Signature_Blocks then the update message is deemed 'Valid' if the first Signature_Block is marked 'Valid' OR the second Signature_Block is marked 'Valid'.)

6. Algorithms and Extensibility

6.1. Algorithm Suite Considerations

Note that there is currently no support for bilateral negotiation (using BGP capabilities) between BGPsec peers to use a particular (digest and signature) algorithm suite. This is because the algorithm suite used by the sender of a BGPsec update message MUST be understood not only by the peer to whom it is directly sending the message, but also by all BGPsec speakers to whom the route advertisement is eventually propagated. Therefore, selection of an algorithm suite cannot be a local matter negotiated by BGP peers, but instead must be coordinated throughout the Internet.

To this end, a mandatory algorithm suites document exists which specifies a mandatory-to-use ’current’ algorithm suite for use by all BGPsec speakers [I-D.ietf-sidr-bgpsec-algs].

It is anticipated that, in the future, the mandatory algorithm suites document will be updated to specify a transition from the ’current’ algorithm suite to a ’new’ algorithm suite. During the period of transition, all BGPsec update messages SHOULD simultaneously use both the ’current’ algorithm suite and the ’new’ algorithm suite. (Note that Section 3 and Section 4 specify how the BGPsec_Path attribute can contain signatures, in parallel, for two algorithm suites.) Once the transition is complete, use of the old ’current’ algorithm will be deprecated, use of the ’new’ algorithm will be mandatory, and a subsequent ’even newer’ algorithm suite may be specified as recommended to implement. Once the transition has successfully been completed in this manner, BGPsec speakers SHOULD include only a single Signature_Block (corresponding to the ’new’ algorithm).

6.2. Considerations for the SKI Size

Depending on the method of generating key identifiers [RFC7093], the size of the SKI in a RPKI router certificate may vary. The SKI field in the BGPsec_Path attribute has a fixed 20 octets size (see Figure 7). If the SKI is longer than 20 octets, then use the leftmost 20 octets of the SKI (excluding the tag and length) [RFC7093]. If the SKI value is shorter than 20 octets, then pad the SKI (excluding the tag and length) to the right (least significant octets) with octets having zero values.

6.3. Extensibility Considerations

This section discusses potential changes to BGPsec that would require substantial changes to the processing of the BGPsec_Path and thus necessitate a new version of BGPsec. Examples of such changes include:

- A new type of signature algorithm that produces signatures of variable length
- A new type of signature algorithm for which the number of signatures in the Signature_Block is not equal to the number of ASes in the Secure_Path (e.g., aggregate signatures)
- Changes to the data that is protected by the BGPsec signatures (e.g., attributes other than the AS path)

In the case that such a change to BGPsec were deemed desirable, it is expected that a subsequent version of BGPsec would be created and that this version of BGPsec would specify a new BGP path attribute, let’s call it BGPsec_Path_Two, which is designed to accommodate the desired changes to BGPsec. In such a case, the mandatory algorithm suites document would be updated to specify algorithm suites appropriate for the new version of BGPsec.

At this point a transition would begin which is analogous to the algorithm transition discussed in Section 6.1. During the transition period all BGPsec speakers SHOULD simultaneously include both the BGPsec_Path attribute and the new BGPsec_Path_Two attribute. Once the transition is complete, the use of BGPsec_Path could then be deprecated, at which point BGPsec speakers should include only the new BGPsec_Path_Two attribute. Such a process could facilitate a transition to a new BGPsec semantics in a backwards compatible fashion.

7. Operations and Management Considerations

Some operations and management issues that are closely relevant to BGPsec protocol specification and its deployment are highlighted here. The Best Current Practices concerning operations and deployment of BGPsec are provided in [I-D.ietf-sidr-bgpsec-ops].

7.1. Capability Negotiation Failure

Section 2.2 describes the negotiation required to establish a BGPsec-capable peering session. Not only must the BGPsec capability be exchanged (and agreed on), but the BGP multiprotocol extension [RFC4760] for the same AFI and the four-byte AS capability [RFC6793] MUST also be exchanged. Failure to properly negotiate a BGPsec session, due to a missing capability, for example, may still result in the exchange of BGP (unsigned) updates. It is RECOMMENDED that an implementation log the failure to properly negotiate a BGPsec session. Also, an implementation MUST have the ability to prevent a BGP session from being established if configured for only BGPsec use.

7.2. Preventing Misuse of pCount=0

A peer that is an Internet Exchange Point (IXP) (i.e. Route Server) with a transparent AS is expected to set pCount=0 in its Secure_Path Segment while forwarding an update to a peer (see Section 4.2). Clearly, such an IXP MUST configure its BGPsec router to set pCount=0 in its Secure_Path Segment. This also means that a BGPsec speaker MUST be configured so that it permits pCount=0 from an IXP peer. Two other cases where pCount is set to zero are in the context AS confederation (see Section 4.3) and AS migration [I-D.ietf-sidr-as-migration]. In these two cases, pCount=0 is set and accepted within the same AS (albeit the AS has two different identities). Note that if a BGPsec speaker does not expect a peer AS to set its pCount=0, and if an update received from that peer violates this, then the update MUST be considered to be in error (see the list of checks in Section 5.2). See Section 8.4 for a discussion of security considerations concerning pCount=0.

7.3. Early Termination of Signature Verification

During the validation of a BGPsec update, route processor performance speedup can be achieved by incorporating the following observations. An update is deemed ’Valid’ if at least one of the Signature_Blocks is marked as ’Valid’ (see Section 5.2). Therefore, if an update contains two Signature_Blocks and the first one verified is found ’Valid’, then the second Signature_Block does not have to be verified. And if the update is chosen for best path, then the BGPsec speaker adds its signature (generated with the respective algorithm) to each of the two Signature_Blocks and forwards the update. Also, a BGPsec update is deemed ’Not Valid’ if at least one signature in each of the Signature_Blocks is invalid. This principle can also be used for route processor workload savings, i.e. the verification for a Signature_Block terminates early when the first invalid signature is encountered.

7.4. Non-Deterministic Signature Algorithms

Many signature algorithms are non-deterministic. That is, many signature algorithms will produce different signatures each time they are run (even when they are signing the same data with the same key). Therefore, if a BGPsec router receives a BGPsec update from a peer and later receives a second BGPsec update message from the same peer for the same prefix with the same Secure_Path and SKIs, the second update MAY differ from the first update in the signature fields (for a non-deterministic signature algorithm). However, the two sets of signature fields will not differ if the sender caches and reuses the previous signature. For a deterministic signature algorithm, the signature fields MUST be identical between the two updates. On the basis of these observations, an implementation MAY incorporate optimizations in update validation processing.

7.5. Private AS Numbers

It is possible that a stub customer of an ISP employs a private AS number. Such a stub customer cannot publish a ROA in the global RPKI for the private AS number and the prefixes that they use. Also, the global RPKI cannot support private AS numbers (i.e. BGPsec speakers in private ASes cannot be issued router certificates in the global RPKI). For interactions between the stub customer (with private AS number) and the ISP, the following two scenarios are possible:

1. The stub customer sends an unsigned BGP update for a prefix to the ISP’s AS. An edge BGPsec speaker in the ISP’s AS may choose to propagate the prefix to its non-BGPsec and BGPsec peers. If so, the ISP’s edge BGPsec speaker MUST strip the AS_PATH with the private AS number, and then (a) re-originate the prefix without any signatures towards its non-BGPsec peer and (b) re-originate the prefix including its own signature towards its BGPsec peer. In both cases (i.e. (a) and (b)), the prefix MUST have a ROA in the global RPKI authorizing the ISP’s AS to originate it.

2. The ISP and the stub customer may use a local RPKI repository (using a mechanism such as described in [I-D.ietf-sidr-slurm]). Then there can be a ROA for the prefix originated by the stub AS, and the eBGP speaker in the stub AS can be a BGPsec speaker having a router certificate, albeit the ROA and router certificate are valid only locally. With this arrangement, the stub AS sends a signed update for the prefix to the ISP’s AS. An edge BGPsec speaker in the ISP’s AS validates the update using RPKI data based the local RPKI view. Further, it may choose to propagate the prefix to its non-BGPsec and BGPsec peers. If so, the ISP’s edge BGPsec speaker MUST strip the Secure_Path and the Signature Segment received from the stub AS with the private AS number, and then (a) re-originate the prefix without any signatures towards its non-BGPsec peer and (b) re-originate the prefix including its own signature towards its BGPsec peer. In both cases (i.e. (a) and (b)), the prefix MUST have a ROA in the global RPKI authorizing the ISP’s AS to originate it.

It is possible that private AS numbers are used in an AS confederation [RFC5065]. BGPsec protocol requires that when a BGPsec update propagates through a confederation, each Member-AS that forwards it to a peer Member-AS MUST sign the update (see Section 4.3). However, the global RPKI cannot support private AS numbers. In order for the BGPsec speakers in Member-ASes with private AS numbers to have digital certificates, there MUST be a mechanism in place in the confederation that allows establishment of a local, customized view of the RPKI, augmenting the global RPKI repository data as needed. Since this mechanism (for augmenting and maintaining a local image of RPKI data) operates locally within an AS or AS confederation, it need not be standard based. However, a standard-based mechanism can be used (see [I-D.ietf-sidr-slurm]). Recall that in order to prevent exposure of the internals of AS confederations, a BGPsec speaker exporting to a non-member removes all intra-confederation Secure_Path Segments and Signatures (see Section 4.3).

7.6. Robustness Considerations for Accessing RPKI Data

The deployment structure, technologies and best practices concerning global RPKI data to reach routers (via local RPKI caches) are described in [RFC6810] [I-D.ietf-sidr-rpki-rtr-rfc6810-bis] [I-D.ietf-sidr-publication] [RFC7115] [I-D.ietf-sidr-bgpsec-ops] [I-D.ietf-sidr-delta-protocol]. For example, serial-number based incremental update mechanisms are used for efficient transfer of just the data records that have changed since last update [RFC6810] [I-D.ietf-sidr-rpki-rtr-rfc6810-bis]. Update notification file is used by relying parties (RPs) to discover whether any changes exist between the state of the global RPKI repository and the RP’s cache [I-D.ietf-sidr-delta-protocol]. The notification describes the location of the files containing the snapshot and incremental deltas which can be used by the RP to synchronize with the repository. Making use of these technologies and best practices results in enabling robustness, efficiency, and better security for the BGPsec routers and RPKI caches in terms of the flow of RPKI data from repositories to RPKI caches to routers. With these mechanisms, it is believed that an attacker wouldn’t be able to meaningfully correlate RPKI data flows with BGPsec RP (or router) actions, thus avoiding attacks that may attempt to determine the set of ASes interacting with an RP via the interactions between the RP and RPKI servers.

7.7. Graceful Restart

During Graceful Restart (GR), restarting and receiving BGPsec speakers MUST follow the procedures specified in [RFC4724] for restarting and receiving BGP speakers, respectively. In particular, the behavior of retaining the forwarding state for the routes in the Loc-RIB [RFC4271] and marking them as stale as well as not differentiating between stale and other information during forwarding will be the same as specified in [RFC4724].

7.8. Robustness of Secret Random Number in ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA) with curve P-256 is used for signing updates in BGPsec [I-D.ietf-sidr-bgpsec-algs]. For ECDSA, it is stated in Section 6.3 of [FIPS186-4] that a new secret random number "k" shall be generated prior to the generation of each digital signature. A high entropy random bit generator (RBG) must be used for generating "k", and any potential bias in the "k" generation algorithm must be mitigated (see methods described in [FIPS186-4] [SP800-90A]).

7.9. Incremental/Partial Deployment Considerations

How will migration from BGP to BGPsec look like? What are the benefits for the first adopters? Initially small groups of contiguous ASes would be doing BGPsec. There would be possibly one or more such groups in different geographic regions of the global Internet. Only the routes originated within each group and propagated within its borders would get the benefits of cryptographic AS path protection. As BGPsec adoption grows, each group grows in size and eventually they join together to form even larger BGPsec capable groups of contiguous ASes. The benefit for early adopters starts with AS path security within the contiguous-AS regions spanned by their respective groups. Over time they would see those contiguous-AS regions grow much larger.

During partial deployment, if an AS in the path doesn’t support BGPsec, then BGP goes back to traditional mode, i.e. BGPsec updates are converted to unsigned updates before forwarding to that AS (see Section 4.4). At this point, the assurance that the update propagated via the sequence of ASes listed is lost. In other words, for the BGPsec routers residing in the ASes starting from the origin AS to the AS before the one not supporting BGPsec, the assurance can be still provided, but not beyond that (for the updates in consideration).

8. Security Considerations

For a discussion of the BGPsec threat model and related security considerations, please see RFC 7132 [RFC7132].

8.1. Security Guarantees

When used in conjunction with Origin Validation (see RFC 6483 [RFC6483] and RFC 6811 [RFC6811]), a BGPsec speaker who receives a valid BGPsec update message, containing a route advertisement for a given prefix, is provided with the following security guarantees:

- The origin AS number corresponds to an autonomous system that has been authorized, in the RPKI, by the IP address space holder to originate route advertisements for the given prefix.
- For each AS in the path, a BGPsec speaker authorized by the holder of the AS number intentionally chose (in accordance with local policy) to propagate the route advertisement to the subsequent AS in the path.

That is, the recipient of a valid BGPsec update message is assured that the update propagated via the sequence of ASes listed in the

Secure_Path portion of the BGPsec_Path attribute. (It should be noted that BGPsec does not offer any guarantee that the data packets would flow along the indicated path; it only guarantees that the BGP update conveying the path indeed propagated along the indicated path.) Furthermore, the recipient is assured that this path terminates in an autonomous system that has been authorized by the IP address space holder as a legitimate destination for traffic to the given prefix.

Note that although BGPsec provides a mechanism for an AS to validate that a received update message has certain security properties, the use of such a mechanism to influence route selection is completely a matter of local policy. Therefore, a BGPsec speaker can make no assumptions about the validity of a route received from an external (eBGP) BGPsec peer. That is, a compliant BGPsec peer may (depending on the local policy of the peer) send update messages that fail the validity test in Section 5. Thus, a BGPsec speaker MUST completely validate all BGPsec update messages received from external peers. (Validation of update messages received from internal peers is a matter of local policy, see Section 5.)

8.2. On the Removal of BGPsec Signatures

There may be cases where a BGPsec speaker deems 'Valid' (as per the validation algorithm in Section 5.2) a BGPsec update message that contains both a 'Valid' and a 'Not Valid' Signature_Block. That is, the update message contains two sets of signatures corresponding to two algorithm suites, and one set of signatures verifies correctly and the other set of signatures fails to verify. In this case, the protocol specifies that a BGPsec speaker choosing to propagate the route advertisement in such an update message MUST add its signature to each of the Signature_Blocks (see Section 4.2). Thus the BGPsec speaker creates a signature using both algorithm suites and creates a new update message that contains both the 'Valid' and the 'Not Valid' set of signatures (from its own vantage point).

To understand the reason for such a design decision, consider the case where the BGPsec speaker receives an update message with both a set of algorithm A signatures which are 'Valid' and a set of algorithm B signatures which are 'Not Valid'. In such a case it is possible (perhaps even likely, depending on the state of the algorithm transition) that some of the BGPsec speaker's peers (or other entities further 'downstream' in the BGP topology) do not support algorithm A. Therefore, if the BGPsec speaker were to remove the 'Not Valid' set of signatures corresponding to algorithm B, such entities would treat the message as though it were unsigned. By including the 'Not Valid' set of signatures when propagating a route advertisement, the BGPsec speaker ensures that 'downstream' entities have as much information as possible to make an informed opinion about the validation status of a BGPsec update.

Note also that during a period of partial BGPsec deployment, a ’downstream’ entity might reasonably treat unsigned messages differently from BGPsec updates that contain a single set of ’Not Valid’ signatures. That is, by removing the set of ’Not Valid’ signatures the BGPsec speaker might actually cause a downstream entity to ’upgrade’ the status of a route advertisement from ’Not Valid’ to unsigned. Finally, note that in the above scenario, the BGPsec speaker might have deemed algorithm A signatures ’Valid’ only because of some issue with RPKI state local to its AS (for example, its AS might not yet have obtained a CRL indicating that a key used to verify an algorithm A signature belongs to a newly revoked certificate). In such a case, it is highly desirable for a downstream entity to treat the update as ’Not Valid’ (due to the revocation) and not as ’unsigned’ (which would happen if the ’Not Valid’ Signature_Blocks were removed enroute).

A similar argument applies to the case where a BGPsec speaker (for some reason such as lack of viable alternatives) selects as its best path (to a given prefix) a route obtained via a ’Not Valid’ BGPsec update message. In such a case, the BGPsec speaker should propagate a signed BGPsec update message, adding its signature to the ’Not Valid’ signatures that already exist. Again, this is to ensure that ’downstream’ entities are able to make an informed decision and not erroneously treat the route as unsigned. It should also be noted that due to possible differences in RPKI data observed at different vantage points in the network, a BGPsec update deemed ’Not Valid’ at an upstream BGPsec speaker may be deemed ’Valid’ by another BGP speaker downstream.

Indeed, when a BGPsec speaker signs an outgoing update message, it is not attesting to a belief that all signatures prior to its are valid. Instead it is merely asserting that:

- The BGPsec speaker received the given route advertisement with the indicated prefix, AFI, SAFI, and Secure_Path; and

- The BGPsec speaker chose to propagate an advertisement for this route to the peer (implicitly) indicated by the ’Target AS Number’.

8.3. Mitigation of Denial of Service Attacks

The BGPsec update validation procedure is a potential target for denial of service attacks against a BGPsec speaker. The mitigation of denial of service attacks that are specific to the BGPsec protocol is considered here.

To mitigate the effectiveness of such denial of service attacks, BGPsec speakers should implement an update validation algorithm that performs expensive checks (e.g., signature verification) after performing less expensive checks (e.g., syntax checks). The validation algorithm specified in Section 5.2 was chosen so as to perform checks which are likely to be expensive after checks that are likely to be inexpensive. However, the relative cost of performing required validation steps may vary between implementations, and thus the algorithm specified in Section 5.2 may not provide the best denial of service protection for all implementations.

Additionally, sending update messages with very long AS paths (and hence a large number of signatures) is a potential mechanism to conduct denial of service attacks. For this reason, it is important that an implementation of the validation algorithm stops attempting to verify signatures as soon as an invalid signature is found. (This ensures that long sequences of invalid signatures cannot be used for denial of service attacks.) Furthermore, implementations can mitigate such attacks by only performing validation on update messages that, if valid, would be selected as the best path. That is, if an update message contains a route that would lose out in best path selection for other reasons (e.g., a very long AS path) then it is not necessary to determine the BGPsec-validity status of the route.

8.4. Additional Security Considerations

The mechanism of setting the pCount field to zero is included in this specification to enable route servers in the control path to participate in BGPsec without increasing the length of the AS path. Two other scenarios where pCount=0 is utilized are in the context AS confederation (see Section 4.3) and AS migration [I-D.ietf-sidr-as-migration]. In these two scenarios, pCount=0 is set and also accepted within the same AS (albeit the AS has two different identities). However, entities other than route servers, confederation ASes or migrating ASes could conceivably use this mechanism (set the pCount to zero) to attract traffic (by reducing the length of the AS path) illegitimately. This risk is largely mitigated if every BGPsec speaker follows the operational guidance in Section 7.2 for configuration for setting pCount=0 and/or accepting pCount=0 from a peer. However, note that a recipient of a BGPsec update message within which an upstream entity two or more hops away has set pCount to zero is unable to verify for themselves whether pCount was set to zero legitimately.

There is a possibility of passing a BGPsec update via tunneling between colluding ASes. For example, say, AS-X does not peer with AS-Y, but colludes with AS-Y, signs and sends a BGPsec update to AS-Y by tunneling. AS-Y can then further sign and propagate the BGPsec update to its peers. It is beyond the scope of the BGPsec protocol to detect this form of malicious behavior. BGPsec is designed to protect messages sent within BGP (i.e. within the control plane) - not when the control plane in bypassed.

A variant of the collusion by tunneling mentioned above can happen in the context of AS confederations. When a BGPsec router (outside of a confederation) is forwarding an update to a Member-AS in the confederation, it signs the update to the public AS number of the confederation and not to the member’s AS number (see Section 4.3). The Member-AS can tunnel the signed update to another Member-AS as received (i.e. without adding a signature). The update can then be propagated using BGPsec to other confederation members or to BGPsec neighbors outside of the confederation. This kind of operation is possible, but no grave security or reachability compromise is feared for the following reasons: (1) The confederation members belong to one organization and strong internal trust is expected; and (2) Recall that the signatures that are internal to the confederation MUST be removed prior to forwarding the update to an outside BGPsec router (see Section 4.3).

BGPsec does not provide protection against attacks at the transport layer. As with any BGP session, an adversary on the path between a BGPsec speaker and its peer is able to perform attacks such as modifying valid BGPsec updates to cause them to fail validation, injecting (unsigned) BGP update messages without BGPsec_Path attributes, injecting BGPsec update messages with BGPsec_Path attributes that fail validation, or causing the peer to tear-down the BGP session. The use of BGPsec does nothing to increase the power of an on-path adversary -- in particular, even an on-path adversary cannot cause a BGPsec speaker to believe a BGPsec-invalid route is valid. However, as with any BGP session, BGPsec sessions SHOULD be protected by appropriate transport security mechanisms (see the Security Considerations section in [RFC4271]).

There is a possibility of replay attacks which are defined as follows. In the context of BGPsec, a replay attack occurs when a malicious BGPsec speaker in the AS path suppresses a prefix withdrawal (implicit or explicit). Further, a replay attack is said to occur also when a malicious BGPsec speaker replays a previously received BGPsec announcement for a prefix that has since been withdrawn. The mitigation strategy for replay attacks involves router certificate rollover; please see [I-D.ietf-sidrops-bgpsec-rollover] for details.

9. IANA Considerations

IANA is requested to register a new BGP capability from Section 2.1 in the BGP Capabilities Code registry’s "IETF Review" range. The description for the new capability is "BGPsec Capability". The reference for the new capability is this document (i.e. the RFC that replaces draft-ietf-sidr-bgpsec-protocol).

IANA is also requested to register a new path attribute from Section 3 in the BGP Path Attributes registry. The code for this new attribute is "BGPsec_Path". The reference for the new attribute is this document (i.e. the RFC that replaces draft-ietf-sidr-bgpsec-protocol).

IANA is requested to define the "BGPsec Capability" registry in the Resource Public Key Infrastructure (RPKI) group. The registry is as shown in Figure 10 with values assigned from Section 2.1:

Bits	Field	Reference
0-3	Version Value = 0x0	[This RFC]
4	Direction (Both possible values 0 and 1 are fully specified by this RFC)	[This RFC]
5-7	Unassigned Value = 000 (in binary)	[This RFC]

Figure 10: IANA registry for BGPsec Capability.

The Direction bit (4th bit) has value either 0 or 1, and both values are fully specified by this document (i.e. the RFC that replaces draft-ietf-sidr-bgpsec-protocol). Future Version values and future values of the Unassigned bits are assigned using the "Standards Action" registration procedures defined in RFC 5226 [RFC5226].

IANA is requested to define the "BGPsec_Path Flags" registry in the RPKI group. The registry is as shown in Figure 11 with one value assigned from Section 3.1:

Flag	Description	Reference
0	Confed_Segment Bit value = 1 means Flag set (indicates Confed_Segment) Bit value = 0 is default	[This RFC]
1-7	Unassigned Value: All 7 bits set to zero	[This RFC]

Figure 11: IANA registry for BGPsec_Path Flags field.

Future values of the Unassigned bits are assigned using the "Standards Action" registration procedures defined in RFC 5226 [RFC5226].

10. Contributors

10.1. Authors

Rob Austein
Dragon Research Labs
sra@hactrn.net

Steven Bellovin
Columbia University
smb@cs.columbia.edu

Randy Bush
Internet Initiative Japan
randy@psg.com

Russ Housley
Vigil Security
housley@vigilsec.com

Matt Lepinski
New College of Florida
mlepinski@ncf.edu

Stephen Kent
BBN Technologies
kent@bbn.com

Warren Kumari

Google
warren@kumari.net

Doug Montgomery
USA National Institute of Standards and Technology
dougm@nist.gov

Kotikalapudi Sriram
USA National Institute of Standards and Technology
kotikalapudi.sriram@nist.gov

Samuel Weiler
W3C/MIT
weiler@csail.mit.edu

10.2. Acknowledgements

The authors would like to thank Michael Baer, Oliver Borchert, David Mandelberg, Mehmet Adalier, Sean Turner, John Scudder, Wes George, Jeff Haas, Keyur Patel, Alvaro Retana, Nevil Brownlee, Matthias Waehlisch, Sandy Murphy, Chris Morrow, Tim Polk, Russ Mundy, Wes Hardaker, Sharon Goldberg, Ed Kern, Doug Maughan, Pradosh Mohapatra, Mark Reynolds, Heather Schiller, Jason Schiller, Ruediger Volk, and David Ward for their review, comments, and suggestions during the course of this work. Thanks are also due to many IESG reviewers whose comments greatly helped improve the clarity, accuracy, and presentation in the document.

11. References

11.1. Normative References

[I-D.ietf-sidr-bgpsec-algs]
Turner, S. and O. Borchert, "BGPsec Algorithms, Key Formats, & Signature Formats", draft-ietf-sidr-bgpsec-algs-18 (work in progress), April 2017.

[I-D.ietf-sidr-bgpsec-pki-profiles]
Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bgpsec-pki-profiles-21 (work in progress), January 2017.

[IANA-AF] "Address Family Numbers", <http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <http://www.rfc-editor.org/info/rfc4724>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <http://www.rfc-editor.org/info/rfc4760>.

[RFC5065] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, DOI 10.17487/RFC5065, August 2007, <http://www.rfc-editor.org/info/rfc5065>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <http://www.rfc-editor.org/info/rfc5492>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <http://www.rfc-editor.org/info/rfc6482>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <http://www.rfc-editor.org/info/rfc6487>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <http://www.rfc-editor.org/info/rfc6793>.

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <http://www.rfc-editor.org/info/rfc7606>.

11.2. Informative References

[Borchert]
Borchert, O. and M. Baer, "Modification request: draft-ietf-sidr-bgpsec-protocol-14", IETF SIDR WG Mailing List message , February 10, 2016, <https://mailarchive.ietf.org/arch/msg/sidr/8B_e4CNxQCUKeZ_AUzsdnn2f5Mu>.

[FIPS186-4]
"FIPS Standards Publication 186-4: Digital Signature Standard", July 2013, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.

[I-D.ietf-sidr-as-migration]
George, W. and S. Murphy, "BGPSec Considerations for AS Migration", draft-ietf-sidr-as-migration-06 (work in progress), December 2016.

[I-D.ietf-sidr-bgpsec-ops]
Bush, R., "BGPsec Operational Considerations", draft-ietf-sidr-bgpsec-ops-16 (work in progress), January 2017.

[I-D.ietf-sidr-delta-protocol]
Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, "RPKI Repository Delta Protocol (RRDP)", draft-ietf-sidr-delta-protocol-08 (work in progress), March 2017.

[I-D.ietf-sidr-publication]
Weiler, S., Sonalker, A., and R. Austein, "A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", draft-ietf-sidr-publication-12 (work in progress), March 2017.

[I-D.ietf-sidr-rpki-rtr-rfc6810-bis]
Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", draft-ietf-sidr-rpki-rtr-rfc6810-bis-09 (work in progress), February 2017.

[I-D.ietf-sidr-slurm]
Mandelberg, D., Ma, D., and T. Bruijnzeels, "Simplified Local internet nUmber Resource Management with the RPKI", draft-ietf-sidr-slurm-04 (work in progress), March 2017.

[I-D.ietf-sidrops-bgpsec-rollover]
Weis, B., Gagliano, R., and K. Patel, "BGPsec Router Certificate Rollover", draft-ietf-sidrops-bgpsec-rollover-00 (work in progress), March 2017.

[RFC6472] Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", BCP 172, RFC 6472, DOI 10.17487/RFC6472, December 2011, <http://www.rfc-editor.org/info/rfc6472>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <http://www.rfc-editor.org/info/rfc6480>.

[RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, DOI 10.17487/RFC6483, February 2012, <http://www.rfc-editor.org/info/rfc6483>.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <http://www.rfc-editor.org/info/rfc6810>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <http://www.rfc-editor.org/info/rfc6811>.

[RFC7093] Turner, S., Kent, S., and J. Manger, "Additional Methods for Generating Key Identifiers Values", RFC 7093, DOI 10.17487/RFC7093, December 2013, <http://www.rfc-editor.org/info/rfc7093>.

[RFC7115] Bush, R., "Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)", BCP 185, RFC 7115, DOI 10.17487/RFC7115, January 2014, <http://www.rfc-editor.org/info/rfc7115>.

[RFC7132] Kent, S. and A. Chi, "Threat Model for BGP Path Security", RFC 7132, DOI 10.17487/RFC7132, February 2014, <http://www.rfc-editor.org/info/rfc7132>.

[SP800-90A]
"NIST 800-90A: Deterministic Random Bit Generator Validation System", October 2015, <http://csrc.nist.gov/groups/STM/cavp/documents/drbg/DRBGVS.pdf>.

Authors’ Addresses

Matthew Lepinski (editor)
NCF
5800 Bay Shore Road
Sarasota FL 34243
USA

Email: mlepinski@ncf.edu

Kotikalapudi Sriram (editor)
NIST
100 Bureau Drive
Gaithersburg MD 20899
USA

Email: kotikalapudi.sriram@nist.gov

Network Working Group S. Bellovin
Internet-Draft Columbia University
Intended status: Standards Track R. Bush
Expires: September 12, 2012 Internet Initiative Japan
D. Ward
Cisco Systems
March 11, 2012

Security Requirements for BGP Path Validation
draft-ietf-sidr-bgpsec-reqs-03

Abstract

This document describes requirements for a future BGP security protocol design to provide cryptographic assurance that the origin AS had the right to announce the prefix and to provide assurance of the AS Path of the announcement.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft will expire on September 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

RPKI-based Origin Validation ([I-D.ietf-sidr-pfx-validate]) provides a measure of resilience to accidental mis-origination of prefixes. But it provides neither cryptographic assurance (announcements are not signed), nor assurance of the AS Path of the announcement.

This document describes requirements to be placed on a BGP security protocol, herein termed BGPsec, intended to rectify these gaps.

The threat model assumed here is documented in [RFC4593] and [I-D.ietf-sidr-bgpsec-threats].

As noted in the threat model, [I-D.ietf-sidr-bgpsec-threats], this work is limited to threats to the BGP protocol. Issues of business relationship confomance, of which routing ’leaks’ are a subset, while important are outside the scope of the working group and therefore this document. It is hoped that these issues will be better understood in the future.

2. Recommended Reading

This document assumes knowledge of the RPKI see [RFC6480], the RPKI Repository Structure, see [RFC6481].

This document assumes ongoing incremental deployment of ROAs, see [RFC6482], the RPKI to Router Protocol, see [I-D.ietf-sidr-rpki-rtr], and RPKI-based Prefix Validation, see [I-D.ietf-sidr-pfx-validate].

And, of course, a knowledge of BGP [RFC4271] is required.

3. General Requirements

The following are general requirements for a BGPsec protocol:

3.1 A BGPsec design must allow the receiver of a BGP announcement to determine, to a strong level of certainty, that the received PATH attribute accurately represents the sequence of eBGP exchanges that propagated the prefix from the origin AS to the receiver.

3.2 A BGPsec design must allow the receiver of an announcement to detect if an AS has added or deleted any AS number other than its own in the path attribute. This includes modification to the number of AS prepends.

3.3 A BGPsec design MUST be amenable to incremental deployment. Any incompatible protocol capabilities MUST be negotiated.

3.4 A BGPsec design MUST provide analysis of the operational considerations for deployment and particularly of incremental deployment, e.g, contiguous islands, non-contiguous islands, universal deployment, etc..

3.5 As cryptographic payloads and memory requirements on routers are likely to increase, a BGPsec design MAY require use of new hardware. I.e. compatibility with current hardware abilities is not a requirement that this document imposes on a solution. As BGPsec will likely not be rolled out for some years, this should not be a major problem.

3.6 A BGPsec design need not prevent attacks on data plane traffic. It need not provide assurance that the data plane even follows the control plane.

3.7 A BGPsec design MUST resist attacks by an enemy who has access to the inter-router link layer, per Section 3.1.1.2 of [RFC4593]. In particular, such a design must provide mechanisms for authentication of all data, including protecting against message insertion, deletion, modification, or replay. Mechanisms that suffice include TCP sessions authenticated with TCP-AO [RFC5925], IPsec [RFC4301], or TLS [RFC5246].

3.8 It is assumed that a BGPsec design will require information about holdings of address space and ASNs, and assertions about binding of address space to ASNs. A BGPsec design MAY make use of a security infrastructure (e.g., a PKI) to distribute such authenticated data.

3.9 [this point should probably be removed. it remains to keep numbering for the moment] If message signing increases message size, the 4096 byte limit on BGP PDU size MAY be removed, see [I-D.ietf-idr-bgp-extended-messages].

3.10 It is entirely OPTIONAL to secure AS SETs and prefix aggregation. The long range solution to this is the deprecation of AS-SETs, see [I-D.ietf-idr-deprecate-as-sets].

3.11 If a BGPsec design uses signed prefixes, given the difficulty of splitting a signed message while preserving the signature, it need NOT handle multiple prefixes in a single UPDATE PDU.

3.12 A BGPsec design MUST enable each BGPsec speaker to configure use of the security mechanism on a per-peer basis.

3.13 A BGPsec design MUST provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment. Message formatting in a fully secured environment MAY be handled in a non-backward compatible manner.

3.14 While the trust level of an NLRI should be determined by the BGPsec protocol, local routing preference and policy MUST then be applied to best path and other decisions. Such mechanisms MUST conform with [I-D.ietf-sidr-ltamgmt].

3.15 A BGPsec design MUST support 'transparent' route servers, meaning that the AS of the route server is not counted in downstream BGP AS-path-length tie-breaking decisions.

3.16 If a BGPsec design makes use of a security infrastructure, that infrastructure SHOULD enable each network operator to select the entities it will trust when authenticating data in the security infrastructure. See, for example, [I-D.ietf-sidr-ltamgmt].

3.17 A BGPsec design MUST NOT require operators to reveal more than is currently revealed in the operational inter-domain routing environment, other than the inclusion of necessary security credentials to allow others to ascertain for themselves the necessary degree of assurance regarding the validity of NLRI received via BGPsec. This includes peering, customer, and provider relationships, an ISP's internal infrastructure, etc. It is understood that some data are revealed to the savvy seeker by BGP, traceroute, etc. today.

3.18 A BGPsec design SHOULD flag security exceptions which are significant enough to be logged. The specific data to be logged are an implementation matter.

3.19 Any routing information database MUST be re-authenticated periodically or in an event-driven manner, especially in response to events such as, for example, PKI updates.

3.20 Any inter-AS use of cryptographic hashes or signatures, MUST provide mechanisms for algorithm agility.

3.21 A BGPsec design SHOULD NOT presume to know the intent of the originator of a NLRI, nor that of any AS on the AS Path.

3.22 A BGP listener SHOULD NOT trust non-BGPsec markings, such as communities, across trust boundaries.

4. BGP UPDATE Security Requirements

The following requirements MUST be met in the processing of BGP UPDATE messages:

4.1 A BGPsec design MUST enable each recipient of an UPDATE to formally validate that the origin AS in the message is authorized to originate a route to the prefix(es) in the message.

4.2 A BGPsec design MUST enable the recipient of an UPDATE to formally determine that the NLRI has traversed the AS path indicated in the UPDATE. Note that this is more stringent than showing that the path is merely not impossible.

4.3 Replay of BGP UPDATE messages need not be completely prevented, but a BGPsec design MUST provide a mechanism to control the window of exposure to replay attacks.

4.4 A BGPsec design SHOULD provide some level of assurance that the origin of a prefix is still ’alive’, i.e. that a monkey in the middle has not withheld a WITHDRAW message or the effects thereof.

4.5 NLRI of the UPDATE message SHOULD be able to be authenticated as the message is processed.

4.6 Normal sanity checks of received announcements MUST be done, e.g. verification that the first element of the AS_PATH list corresponds to the locally configured AS of the peer from which the UPDATE was received.

4.7 The output of a router applying BGPsec to a received signed UPDATE MUST be either unequivocal and conform to a fully specified state in the design.

5. IANA Considerations

This document asks nothing of the IANA.

6. Security Considerations

The data plane may not follow the control plane.

Security for subscriber traffic is outside the scope of this document, and of BGP security in general. IETF standards for payload data security should be employed. While adoption of BGP security measures may ameliorate some classes of attacks on traffic, these measures are not a substitute for use of subscriber-based security.

7. Acknowledgments

The author wishes to thank the authors of [I-D.ietf-rpsec-bgpsecrec] from whom we liberally stole, Russ Housley, Geoff Huston, Steve Kent, Sandy Murphy, John Scudder, Sam Weiler, and a number of others.

8. References

8.1. Normative References

[I-D.ietf-sidr-bgpsec-threats]
Kent, S. and A. Chi, "Threat Model for BGP Path Security", draft-ietf-sidr-bgpsec-threats-02 (work in progress), February 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4593] Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

8.2. Informative References

[I-D.ietf-idr-bgp-extended-messages]
Patel, K. and R. Bush, "Extended Message support for BGP", draft-ietf-idr-bgp-extended-messages-02 (work in progress), January 2012.

[I-D.ietf-idr-deprecate-as-sets]
Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", draft-ietf-idr-deprecate-as-sets-06 (work in progress), October 2011.

[I-D.ietf-rpsec-bgpsecrec]
Christian, B. and T. Tauber, "BGP Security Requirements", draft-ietf-rpsec-bgpsecrec-10 (work in progress), November 2008.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-04 (work in progress), December 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-03 (work in progress), October 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-26 (work in progress), February 2012.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

Authors' Addresses

Steven M. Bellovin
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, New York 10027
US

Phone: +1 212 939 7149
Email: bellovin@acm.org

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

David Ward
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dward@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 15, 2015

S. Bellovin
Columbia University
R. Bush
Internet Initiative Japan
D. Ward
Cisco Systems

July 14, 2014

Security Requirements for BGP Path Validation
draft-ietf-sidr-bgpsec-reqs-12

Abstract

This document describes requirements for a BGP security protocol design to provide cryptographic assurance that the origin AS (Autonomous System) had the right to announce the prefix and to provide assurance of the AS Path of the announcement.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Resource Public Key Infrastructure (RPKI)-based Origin Validation, [RFC6811], provides a measure of resilience to accidental mis-origination of prefixes. But it provides neither cryptographic assurance (announcements are not signed), nor assurance of the AS Path of the announcement.

This document describes requirements to be placed on a BGP security protocol, herein termed BGPsec, intended to rectify these gaps.

The threat model assumed here is documented in [RFC4593] and [RFC7132].

As noted in the threat model, [RFC7132], this work is limited to threats to the BGP protocol. Issues of business relationship conformance, while quite important to operators, are not security issues per se, and are outside the scope of this document. It is hoped that these issues will be better understood in the future.

2. Recommended Reading

This document assumes knowledge of the RPKI see [RFC6480], the RPKI Repository Structure, see [RFC6481].

This document assumes ongoing incremental deployment of ROAs, see [RFC6482], the RPKI to Router Protocol, see [RFC6810], and RPKI-based Prefix Validation, see [RFC6811].

And, of course, a knowledge of BGP [RFC4271] is required.

3. General Requirements

The following are general requirements for a BGPsec protocol:

3.1 A BGPsec design MUST allow the receiver of a BGP announcement to determine, to a strong level of certainty, that the originating AS in the received PATH attribute possessed the authority to announce the prefix.

3.2 A BGPsec design MUST allow the receiver of a BGP announcement to determine, to a strong level of certainty, that the received PATH attribute accurately represents the sequence of eBGP exchanges that propagated the prefix from the origin AS to the receiver, particularly if an AS has added or deleted any AS number other than its own in the path attribute. This includes modification to the number of AS prepends.

3.3 BGP attributes other than the AS_PATH are used only locally, or have meaning only between immediate neighbors, may be modified by intermediate systems, and figure less prominently in the decision process. Consequently, it is not appropriate to try to protect such attributes in a BGPsec design.

3.4 A BGPsec design MUST be amenable to incremental deployment. This implies that incompatible protocol capabilities MUST be negotiated.

3.5 A BGPsec design MUST provide analysis of the operational considerations for deployment and particularly of incremental deployment, e.g, contiguous islands, non-contiguous islands, universal deployment, etc.

3.6 As proofs of possession and authentication may require cryptographic payloads and/or storage and computation, likely increasing processing and memory requirements on routers, a BGPsec design MAY require use of new hardware. I.e., compatibility with current hardware abilities is not a requirement that this document imposes on a solution.

3.7 A BGPsec design need not prevent attacks on data plane traffic. It need not provide assurance that the data plane even follows the control plane.

3.8 A BGPsec design MUST resist attacks by an enemy who has access to the inter-router link layer, per Section 3.1.1.2 of [RFC4593]. In particular, such a design MUST provide mechanisms for authentication of all data, including protecting against message insertion, deletion, modification, or replay. Mechanisms that suffice include TCP sessions authenticated with TCP-AO [RFC5925], IPsec [RFC4301], or TLS [RFC5246].

3.9 It is assumed that a BGPsec design will require information about holdings of address space and ASNs (AS Numbers), and assertions about binding of address space to ASNs. A BGPsec design MAY make use of a security infrastructure (e.g., a PKI) to distribute such authenticated data.

3.10 It is entirely OPTIONAL to secure AS SETs and prefix aggregation. The long range solution to this is the deprecation of AS_SETs, see [RFC6472].

3.11 If a BGPsec design uses signed prefixes, given the difficulty of splitting a signed message while preserving the signature, it need not handle multiple prefixes in a single UPDATE PDU.

3.12 A BGPsec design MUST enable each BGPsec speaker to configure use of the security mechanism on a per-peer basis.

3.13 A BGPsec design MUST provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment. Message formatting in a fully secured environment MAY be handled in a non-backward compatible manner.

3.14 While the formal validity of a routing announcement should be determined by the BGPsec protocol, local routing policy MUST be the final arbiter of best path and other routing decisions.

3.15 A BGPsec design MUST support ’transparent’ route servers, meaning that the AS of the route server is not counted in downstream BGP AS-path-length tie-breaking decisions.

3.16 A BGPsec design MUST support AS aliasing. This technique is not well-defined or universally implemented, but is being documented in [I-D.ga-idr-as-migration]. A BGPsec design SHOULD accommodate AS ’migration’ techniques such as common proprietary and non-standard methods which allow a router to have two AS identities, without lengthening the effective AS Path.

3.17 If a BGPsec design makes use of a security infrastructure, that infrastructure SHOULD enable each network operator to select the entities it will trust when authenticating data in the security infrastructure. See, for example, [I-D.ietf-sidr-lta-use-cases].

3.18 A BGPsec design MUST NOT require operators to reveal more than is currently revealed in the operational inter-domain routing environment, other than the inclusion of necessary security credentials to allow others to ascertain for themselves the necessary degree of assurance regarding the validity of NLRI received via BGPsec. This includes peering, customer/provider relationships, an ISP’s internal infrastructure, etc. It is understood that some data are revealed to the savvy seeker by BGP, traceroute, etc. today.

3.19 A BGPsec design MUST signal (logging, SNMP, ...) security exceptions which are significant to the operator. The specific data to be signaled are an implementation matter.

3.20 Any routing information database MUST be re-authenticated periodically or in an event-driven manner, especially in response to events such as, for example, PKI updates.

3.21 Any inter-AS use of cryptographic hashes or signatures, MUST provide mechanisms for algorithm agility. For a discussion, see [I-D.iab-crypto-alg-agility].

3.22 A BGPsec design SHOULD NOT presume to know the intent of the originator of a NLRI, nor that of any AS on the AS Path, other than that they intended to pass it to the next AS in the Path.

3.23 A BGPsec listener SHOULD NOT trust non-BGPsec markings, such as communities, across trust boundaries.

4. BGP UPDATE Security Requirements

The following requirements MUST be met in the processing of BGP UPDATE messages:

4.1 A BGPsec design MUST enable each recipient of an UPDATE to formally validate that the origin AS in the message is authorized to originate a route to the prefix(es) in the message.

4.2 A BGPsec design MUST enable the recipient of an UPDATE to formally determine that the NLRI has traversed the AS path indicated in the UPDATE. Note that this is more stringent than showing that the path is merely not impossible.

4.3 Replay of BGP UPDATE messages need not be completely prevented, but a BGPsec design SHOULD provide a mechanism to control the window of exposure to replay attacks.

4.4 A BGPsec design SHOULD provide some level of assurance that the origin of a prefix is still 'alive', i.e., that a monkey in the middle has not withheld a WITHDRAW message or the effects thereof.

4.5 The AS Path of an UPDATE message SHOULD be able to be authenticated as the message is processed.

4.6 Normal sanity checks of received announcements MUST be done, e.g., verification that the first element of the AS_PATH list corresponds to the locally configured AS of the peer from which the UPDATE was received.

4.7 The output of a router applying BGPsec validation to a received UPDATE MUST be unequivocal and conform to a fully specified state in the design.

5. IANA Considerations

This document asks nothing of the IANA.

6. Security Considerations

If an external "security infrastructure" is used, as mentioned in Paragraph 9 and Paragraph 17 above, the authenticity and integrity of the data of such an infrastructure MUST be assured. And the integrity of those data MUST be assured when they are used by BGPsec, e.g., in transport.

The requirement of backward compatibility to BGP4 may open an avenue to downgrade attacks.

The data plane might not follow the path signaled by the control plane.

Security for subscriber traffic is outside the scope of this document, and of BGP security in general. IETF standards for payload data security should be employed. While adoption of BGP security measures may ameliorate some classes of attacks on traffic, these measures are not a substitute for use of subscriber-based security.

7. Acknowledgments

The authors wishe to thank the authors of [I-D.ietf-rpsec-bgpsecrec] from whom we liberally stole, Roque Gagliano, Russ Housley, Geoff Huston, Steve Kent, Sandy Murphy, Eric Osterweil, John Scudder, Kotikalapudi Sriram, Sam Weiler, and a number of others.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4593] Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC7132] Kent, S. and A. Chi, "Threat Model for BGP Path Security", RFC 7132, February 2014.

8.2. Informative References

[I-D.ga-idr-as-migration]
George, W. and S. Amante, "Autonomous System (AS) Migration Features and Their Effects on the BGP AS_PATH Attribute", draft-ga-idr-as-migration-01 (work in progress), February 2013.

[I-D.iab-crypto-alg-agility]
Housley, R., "Guidelines for Cryptographic Algorithm Agility", draft-iab-crypto-alg-agility-01 (work in progress), June 2014.

[I-D.ietf-rpsec-bgpsecrec]
Christian, B. and T. Tauber, "BGP Security Requirements", draft-ietf-rpsec-bgpsecrec-10 (work in progress), November 2008.

[I-D.ietf-sidr-lta-use-cases]
Bush, R., "RPKI Local Trust Anchor Use Cases", draft-ietf-sidr-lta-use-cases-00 (work in progress), February 2014.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6472] Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", BCP 172, RFC 6472, December 2011.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, January 2013.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, January 2013.

Authors’ Addresses

Steven M. Bellovin
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, New York 10027
USA

Phone: +1 212 939 7149
Email: bellovin@acm.org

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
USA

Email: randy@psg.com

David Ward
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dward@cisco.com

Secure Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2012
February 22, 2012

S. Kent
A. Chi
BBN

Threat Model for BGP Path Security
draft-ietf-sidr-bgpsec-threats-02

Abstract

This document describes a threat model for BGP path security (BGPSEC). It assumes the context established by the SIDR WG charter, as of April 19, 2011. The charter established two goals for the SIDR work:

- Enabling an AS to verify the authorization of an origin AS to originate a specified set of prefixes
- Enabling an AS to verify that the AS-PATH represented in a route matches the path travelled by the NLRI for the route

The charter further mandates that SIDR build upon the Resource Public Key Infrastructure (RPKI), the first product of the WG. Consistent with the charter, this threat model includes an analysis of the RPKI, and focuses on the ability of an AS to verify the authenticity of the AS path info received in a BGP update.

The model assumes that BGP path security is achieved through the application of digital signatures to AS_Path Info. The document characterizes classes of potential adversaries that are considered to be threats, and examines classes of attacks that might be launched against BGPSEC. It concludes with brief discussion of residual vulnerabilities.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the security context in which BGPSEC is intended to operate. It discusses classes of potential adversaries that are considered to be threats, and classes of attacks that might be launched against BGPSEC. Because BGPSEC depends on the Resource Public Key Infrastructure (RPKI) [RFC6480], threats and attacks against the RPKI are included. This model also takes into consideration classes of attacks that are enabled by the use of BGPSEC (based on the current BGPSEC design.)

The motivation for developing BGPSEC, i.e., residual security concerns for BGP, is well described in several documents, including "BGP Security Vulnerabilities Analysis" [RFC4272] and "Design and Analysis of the Secure Border Gateway Protocol (S-BGP)" [Kent2000]. All of these papers note that BGP does not include mechanisms that allow an Autonomous System (AS) to verify the legitimacy and authenticity of BGP route advertisements. (BGP now mandates support for mechanisms to secure peer-peer communication, i.e., for the links that connect BGP routers. There are several secure protocol options to addresses this security concern, e.g., IPsec [RFC4301] and TCP-AO [RFC5925]. This document briefly notes the need to address this aspect of BGP security, but focuses on application layer BGP security issues that are addressed by BGPSEC.)

RFC 4272 [RFC4272] succinctly notes:

> BGP speakers themselves can inject bogus routing information, either by masquerading as any other legitimate BGP speaker, or by distributing unauthorized routing information as themselves. Historically, misconfigured and faulty routers have been responsible for widespread disruptions in the Internet. The legitimate BGP peers have the context and information to produce believable, yet bogus, routing information, and therefore have the opportunity to cause great damage. The cryptographic protections of [TCPMD5] and operational protections cannot exclude the bogus information arising from a legitimate peer. The risk of disruptions caused by legitimate BGP speakers is real and cannot be ignored.

BGPSEC is intended to address the concerns cited above, to provide significantly improved path security, building upon the secure route origination foundation offered by use of the RPKI. Specifically, the RPKI enables relying parties (RPs) to determine if the origin AS for a path was authorized to advertise the prefix contained in a BGP update message. This security feature is enabled by the use of two types of digitally signed data: a PKI [RFC6487] that associates one or more prefixes with the public key(s) of an address space holder, and Route Origination Authorizations (ROAs) [RFC6482] that allows a prefix holder to specify the AS(es) that are authorized to originate routes for a prefix.

The security model adopted for BGPSEC does not assume an "oracle" that can see all of the BGP inputs and outputs associated with every AS or every BGP router. Instead, the model is based on a local notion of what constitutes legitimate, authorized behavior by the BGP routers associated with an AS. This is an AS-centric model of secure operation, consistent with the AS-centric model that BGP employs for routing. This model forms the basis for the discussion that follows.

This document begins with a brief set of definitions relevant to the subsequent sections. It then discusses classes of adversaries that are perceived as viable threats against routing in the public Internet. It continues to explore a range of attacks that might be effected by these adversaries, against both path security and the infrastructure upon which BGPSEC relies. It concludes with a brief review of residual vulnerabilities.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following security and routing terminology definitions are employed in this document.

Adversary - An adversary is an entity (e.g., a person or an organization) perceived as malicious, relative to the security policy of a system. The decision to characterize an entity as an adversary is made by those responsible for the security of a system. Often one describes classes of adversaries with similar capabilities or motivations, rather than specific individuals or organizations.

Attack - An attack is an action that attempts to violate the security policy of a system, e.g., by exploiting a vulnerability. There is often a many to one mapping of attacks to vulnerabilities, because many different attacks may be used to exploit a vulnerability.

Autonomous System (AS) - An AS is a set of one or more IP networks operated by a single administrative entity.

AS Number (ASN) - An ASN is a 2 or 4 byte number issued by a registry to identify an AS in BGP.

Certification Authority (CA) - An entity that issues digital certificates (e.g., X.509 certificates) and vouches for the binding between the data items in a certificate.

Countermeasure - A countermeasure is a procedure or technique that thwarts an attack, preventing it from being successful. Often countermeasures are specific to attacks or classes of attacks.

Border Gateway Protocol (BGP) - A path vector protocol used to convey "reachability" information among autonomous systems, in support of inter-domain routing.

False (Route) Origination - If a network operator originates a route for a prefix that the network operator does not hold (and that it has not been authorized to originate by the prefix holder, this is termed false route origination.

Internet Service Provider (ISP) - An organization managing (and, typically, selling,) Internet services to other organizations or individuals.

Internet Number Resources (INRs) - IPv4 or IPv6 address space and ASNs

Internet Registry - An organization that manages the allocation or distribution of INRs. This encompasses the Internet Assigned Number Authority (IANA), Regional Internet Registries (RIRs), National Internet Registries (NIRs), and Local Internet Registries (LIRs, network operators).

Man in the Middle (MITM) - A MITM is an entity that is able to examine and modify traffic between two (or more) parties on a communication path.

NOC (Network Operations Center) - A network operator employs a set equipment and a staff to manage a network, typically on a 24/7 basis. The equipment and staff are often referred to as the NOC for the network.

Prefix - A prefix is an IP address and a mask used to specify a set of addresses that are grouped together for purposes of routing.

Public Key Infrastructure (PKI) - A PKI is a collection of hardware, software, people, policies, and procedures used to create, manage, distribute, store, and revoke digital certificates.

Relying Parties (RPs) - An RP is an entity that makes use of signed products from a PKI, i.e., relies on signed data that is verified using certificates, and CRLs from a PKI.

RPKI Repository System - The RPKI repository system consists of a distributed set of loosely synchronized databases.

Resource PKI (RPKI) - A PKI operated by the entities that manage INRs, and that issues X509 certificates (and CRLs) that attest to the holdings of INRs.

RPKI Signed Object - An RPKI signed object is a Cryptographic Message Syntax (CMS)-encapsulated data object complying with the format and semantics defined in [RFC6488].

Route - In the Internet, a route is a prefix and an associated sequence of ASNs that indicates a path via which traffic destined for the prefix can be directed. (The route includes the origin AS.)

Route leak - A route leak is said to occur when AS-A advertises routes that it has received from an AS-B to AS-A’s neighbors, but AS-A is not viewed as a transit provider for the prefixes in the route.

Threat - A threat is a motivated, capable adversary. An adversary that is not motivated to launch an attack is not a threat. An adversary that is motivated but not capable of launching an attack also is not a threat.

Vulnerability - A vulnerability is a flaw or weakness in a system’s design, implementation, or operation and management that could be exploited to violate the security policy of a system.

3. Threat Characterization

The following classes of threats are addressed in this document.

Network Operators - A network operator may be a threat. A network operator may be motivated to cause BGP routers it controls to emit update messages with inaccurate routing info, e.g. to cause traffic to flow via paths that are economically advantageous for the operator. Such updates might cause traffic to flow via paths that would otherwise be rejected as less advantageous by other network operators. Because a network operator controls the BGP routers in its network, it is in a position to modify their operation in arbitrary ways. Routers managed by a network operator are vehicles for mounting MITM attacks on both control and data plane traffic. If a network operator participates in the RPKI, it will have at least CA resource certificate and may be able to generate an arbitrary number of subordinate CA certificates and ROAs. It will be authorized to populate (and may even host) its own repository publication point. If it implements BGPSEC, it will have the ability to issue certificates for its routers, and to sign updates in a fashion that will be recognized by BGPSEC-enabled neighbors.

Hackers - Hackers are considered a threat. A hacker might assume control of network management computers and routers controlled by network operators, including network operators that implement BGPSEC. In such cases, hackers would be able to act as a rogue network operators (see above). It is assumed that hackers generally do not have the capability to effect MITM attacks on most links between networks (links used to transmit BGP and subscriber traffic). A hacker might be recruited, without his/her knowledge, by criminals or by nations, to act on their behalf. Hackers may be motivated by a desire for "bragging rights" or for profit.

Criminals - Criminals may be a threat. Criminals might persuade (via threats or extortion) a network operator to act as a rogue network operator (see above), and thus be able to effect a wide range of attacks. Criminals might persuade the staff of a telecommunications provider to enable MITM attacks on links between routers. Motivations for criminals may include the ability to extort money from network operators or network operator clients, e.g., by adversely affecting routing for these network operators or their clients. Criminals also may wish to manipulate routing to conceal the sources of spam, DoS attacks, or other criminal activities.

Registries - Any registry in the RPKI could be a threat. Staff at the registry are capable of manipulating repository content or mismanaging the RPKI certificates that they issue. These actions could adversely affect a network operator or a client of a network operator. The staff could be motivated to do this based on political pressure from the nation in which the registry operates (see below) or due to criminal influence (see above).

Nations - A nation may be a threat. A nation may control one or more network operators that operate in the nation, and thus can cause them to act as rogue network operators. A nation may have a technical active wiretapping capability (e.g., within its territory) that enables it to effect MITM attacks on inter-network traffic. (This capability may be facilitated by control or influence over a telecommunications provider operating within the nation.) It may have an ability to attack and take control of routers or management network computers of network operators in other countries. A nation may control a registry (e.g., an RIR) that operates within its territory, and might force that registry to act in a rogue capacity. National threat motivations include the desire to control the flow of traffic to/from the nation or to divert traffic destined for other nations (for passive or active wiretapping, including DoS).

4. Attack Characterization

This section describes classes of attacks that may be effected against Internet routing (relative to the context described in Section 1). Attacks are classified based on the target of the attack, as an element of the routing system, or the routing security infrastructure on which BGPSEC relies. In general, attacks of interest are ones that attempt to violate the integrity or authenticity of BGP traffic, or which violate the authorizations associated with entities participating in the RPKI. Attacks that violate the implied confidentiality of routing traffic are not considered significant (see Section 4.1 below).

4.1. Active wiretapping of links between routers

An adversary may attack the links that connect BGP routers. Passive attacks are not considered, because it is assumed that most of the info carried by BGP will otherwise be accessible to adversaries. Several classes of adversaries are assumed to be capable of MITM effecting attacks against the control plane traffic. MITM attacks may be directed against BGP, BGPSEC, or against TCP or IP. Such attacks include replay of selected BGP messages, selective modification of BGP messages, and DoS attacks against BGP routers.

4.2. Attacks on a BGP router

An adversary may attack a BGP router, whether it implements BGPSEC or not. Any adversary that controls routers legitimately, or that can assume control of a router, is assumed to be able to effect the types of attacks described below. Note that any router behavior that can be ascribed to a local routing policy decision is not considered to be an attack. This is because such behavior could be explained as a result of local policy settings, and thus is beyond the scope of what BGPSEC can detect as unauthorized behavior. Thus, for example, a router may fail to propagate some or all route withdrawals or effect "route leaks". (These behaviors are not precluded by the specification for BGP, and might be the result of a local policy that is not publicly disclosed. As a result, they are not considered attacks. See Section 5 for additional discussion.)

Attacks on a router are active wiretapping attacks (in the most general sense) that manipulate (forge, tamper with, or suppress) data contained in BGP updates. The list below illustrates attacks of this type.

AS Insertion: A router might insert one or more ASNs, other than its own ASN, into an update message. This violates the BGP spec and thus is considered an attack.

False (Route) Origination: A router might originate a route for a prefix, when the AS that the router represents is not authorized to originate routes for that prefix. This is an attack.

Secure Path Downgrade: A router might remove signatures from a BGPSEC update that it receives, when forwarding this update to a BGPSEC-enabled neighbor. This behavior violates the BGPSEC spec and thus is considered an attack.

Invalid Signature Insertion: A router might emit a signed update with a "bad" signature, i.e., a signature that cannot be validated by other BGPSEC routers. This might be an intentional act, or it might occur due to use of a revoked or expired certificate, a computational error, or a syntactic error. Such behavior violates the BGPSEC spec and thus is considered an attack.

Stale Path Announcement: An announcement may be propagated with an origination signature segment that has expired. This behavior violates the BGPSEC spec and is considered a possible replay attack.

Premature Path Announcement Expiration: A router might emit a signed update with an origin expiry time that is very short. Unless the BGPSEC protocol specification mandates a minimum expiry time, this is not an attack. However, if such a time is mandates, this behavior becomes an attack. BGP speakers along a path generally cannot determine if an expiry time is "suspiciously short" since they cannot know how long a route may have been held by an earlier AS, prior to being released. Thus only an immediate neighbor of a route originator could be expected to detect this type of attack.

MITM Attack: A cryptographic key used for point-to-point security (e.g., TCP-AO, TLS, or IPsec) between two BGP routers might be compromised (e.g., by extraction from a router). This would enable an adversary to effect MITM attacks on the link(s) where the key is used. Use of specific security mechanisms to protect inter-router links between ASes is outside the scope of BGPSEC.

Compromised Router Private Key: The private key associated with an RPKI EE certificate issued to a router might be compromised by an attack against the router. An adversary with access to this key would be able to generate updates that appear to have passed through the AS that this router represents. Such updates might be in injected on a link between the compromised router and its neighbors, if that link is accessible to the adversary. If the adversary controls another network, it could use this key to forge signatures that appear to come from the AS or router(s) in question, with some contraints. So, for example, an adversary that controls another AS could use a compromised router key to issue signed routes that include the targeted AS/router, with limits. (Neighbors of the adversary’s AS ought not accept a route that purports to emanate directly from the targeted AS. So, an adversary can take a legitimate route that passes through the compromised AS, add itself as the next hop, and then forward the resulting route to neighbors.)

Replay Attack: A BGPSEC-protected update may be signed and announced, and later withdrawn. An adversary controlling intermediate routers could fail to propagate the withdrawal, and instead re-announce (i.e., replay) a previous announcement (that has not yet expired). BGP is already vulnerable to behavior of this sort; re-announcement cannot be characterized as an attack, under the assumptions upon which this mode is based (i.e., no oracle).

4.3. Attacks on network operator management computers (non-CA computers)

An adversary may choose to attack computers used by a network operator to manage its network, especially its routers. Such attacks might be effected by an adversary that has compromised the security of these computers. This might be effected via remote attacks, extortion of selected network operations staff, etc. If an adversary compromises NOC computers, it can execute any management function that authorized network operations staff would have performed. Thus the adversary could modify local routing policy to change preferences, to black-hole certain routes, etc. This type of behavior cannot be externally detected as an attack. Externally, this appears as a form of rogue network operator behavior.

If a network operator participates in the RPKI, an adversary could manipulate the RP tools that extract data from the RPKI, causing the output of these tools to be corrupted in various ways. For example, an attack of this sort could cause the network operator to view valid routes as not validated, which could alter its routing behavior.

If an adversary invoked the tool used to manage the repository publication point for this network operator, it could delete any objects stored there (certificates, CRLs, manifests, ROAs, or subordinate CA certificates). This could affect the routing status of entities that have allocations/assignments from this network operator (e.g., by deleting their CA certificates).

An adversary could invoke the tool used to request certificate revocation, causing router certificates, ROAs, or subordinate CA certificates to be revoked. An attack of this sort could affect not only this network operator, but also any network operators that receive allocations/assignments from it, e.g., because their CA certificates were revoked.

If a network operator is BGPSEC-enabled, an attack of this sort could cause the affected network operator to be viewed as not BGPSEC-enabled, possibly making routes it emits be less preferred by other network operators.

If an adversary invoked a tool used to request ROAs, it could effectively re-allocate some of the prefixes allocated/assigned to the network operator (e.g., by modifying the origin AS in ROAs). This might cause other BGPSEC-enabled networks to view the affected network as no longer originating routes for these prefixes. Multi-homed subscribers of this network operator who received an allocation from the network operator might find their traffic was now routed via other connections.

If the network operator is BGPSEC-enabled, and the adversary invoked a tool used to request certificates, it could replace valid certificates for routers with ones that might be rejected by BGPSEC-enabled neighbors.

4.4. Attacks on a repository publication point

A critical element of the RPKI is the repository system. An adversary might attack a repository, or a publication point within a repository, to adversely affect routing.

This section considers only those attacks that can be launched by any adversary who controls a computer hosting one or more repository publication points, without access to the cryptographic keys needed to generate valid RPKI signed products. Such attacks might be effected by an inside or an external threat. Because all repository objects are digitally signed, attacks of this sort translate into DoS attacks against the RPKI RPs. There are a few distinct forms of such attacks, as described below.

Note first that the RPKI calls for RPs to cache the data they acquire and verify from the repository system. Attacks that delete signed products, that insert products with "bad" signatures, that tamper with object signatures, or that replace newer objects with older (valid) ones, can be detected by RPs (with a few exceptions). RPs are expected to make use of local caches. If repository publication points are unavailable or the retrieved data is corrupted, an RP can revert to using the cached data. This behavior helps insulate RPs from the immediate effects of DoS attacks on publication points.

Each RPKI data object has an associated date at which it expires, or is considered stale. (Certificates expire, CRLs become stale.) When an RP uses cached data it is a local decision how to deal with stale or expired data. It is common in PKIs to make use of stale certificate revocation status data, when fresher data is not available. Use of expired certificates is less common, although not unknown. Each RP will decide, locally, whether to continue to make use of or ignore cached RPKI objects that are stale or expired.

If an adversary inserts an object into a publication point, and the object has a "bad" signature, the object will not be accepted and used by RPs.

If an adversary modifies any signed product at a publication point, the signature on the product will fail, causing RPs to not accept it. This is equivalent to deleting the object, in many respects.

If an adversary deletes one or more CA certificates, ROAs or the CRL for a publication point, the manifest for that publication point will allow an RP to detect this attack. (The RP would be very unhappy if there is no CRL for the CA instance anyway.) An RP can continue to use the last valid instance of the deleted object as a local policy option), thus minimizing the impact of such an attack.

If an adversary deletes a manifest (and does not replace it with an older instance), that is detectable by RPs. Such behavior should result in the CA (or publication point maintainer) being notified of the problem. An RP can continue to use the last valid instance of the deleted manifest (a local policy option), thus minimizing the impact of such an attack.

If an adversary deletes newly added CA certificates or ROAs, and replaces the current manifest with the previous manifest, the manifest (and the CRL that it matches) will be "stale" (see [RFC6486]). This alerts an RP that there may be a problem, and, hopefully, the entity responsible for the publication point will be asked to remedy the problem (e.g., republish the missing CA certificates and/or ROAs). An RP cannot know the content of the new certificates or ROAs that are not present, but it can continue to use what it has cached. An attack of this sort will, at least temporarily, cause RPs to be un aware of the newly published objects. INRs associated with these objects will be treated as unauthenticated.

If a CA revokes a CA certificate or a ROA (via deleting the corresponding EE certificate), and the adversary tries to reinstate that CA certificate or ROA, the adversary would have to rollback the CRL and the manifest to undo this action by the CA. As above, this would make the CRL and manifest stale, and this is detectable by RPs. An RP cannot know which CA certificates or ROAs were deleted. Depending on local policy, the RP might use the cached instances of the affected objects, and thus be tricked into making decisions based on these revoked objects. Here too the hope is that the CA will be notified of the problem (by RPs) and will remedy the error.

In the attack scenarios above, when a CRL or manifest is described as stale, this means that the next issue date for the CRL or manifest has passed. Until the next issue date, an RP will not be detect the attack. Thus it behooves CAs to select CRL/manifest lifetimes (the two are linked) that represent an acceptable tradeoff between risk and operational burdens.

Attacks effected by adversaries that are legitimate managers of publication points can have much greater effects, and are discussed below under attacks on or by CAs.

4.5. Attacks on an RPKI CA

Every entity to which INRs have been allocated/assigned is a CA in the RPKI. Each CA is nominally responsible for managing the repository publication point for the set of signed products that it generates. (An INR holder may choose to outsource the operation of the RPKI CA function, and the associated publication point. In such cases, the organization operating on behalf of the INR holder becomes the CA, from an operational and security perspective. The following discussion does not distinguish such outsourced CA operations.)

Note that attacks attributable to a CA may be the result of malice by the CA (i.e., the CA is the adversary) or they may result from a compromise of the CA.

All of adversaries listed in Section 2 are presumed to be capable of launching attacks against the computers used to perform CA functions. Some adversaries might effect an attack on a CA by violating personnel or physical security controls as well. The distinction between CA as adversary vs. CA as an attack victim is important. Only in the latter case should one expect the CA to remedy problems caused by a attack once the attack has been detected. (If a CA does not take such action, the effects are the same as if the CA is an adversary.)

Note that most of the attacks described below do not require disclosure of a CA’s private key to an adversary. If the adversary can gain control of the computer used to issue certificates, it can effect these attacks, even though the private key for the CA remains "secure" (i.e., not disclosed to unauthorized parties). However, if the CA is not the adversary, and if the CA's private key is not compromised, then recovery from these attacks is much easier. This motivates use of hardware security modules to protect CA keys, at least for higher tiers in the RPKI.

An attack by a CA can result in revocation or replacement of any of the certificates that the CA has issued. Revocation of a certificate should cause RPs to delete the (formerly) valid certificate (and associated signed object, in the case of a revoked EE certificate) that they have cached. This would cause repository objects (e.g., CA certificates and ROAs) that are verified under that certificate to be considered invalid, transitively. As a result, RPs would not consider as valid any ROAs or BGPSEC-signed updates based on these certificates, which would make routes dependent on them to be less preferred. Because a CA that revokes a certificate is authorized to do so, this sort of attack cannot be detected, intrinsically, by most RPs. However, the entities affected by the revocation or replacement of CA certificates can be expected to detect the attack and contact the CA to effect remediation. If the CA was not the adversary, it should be able to issue new certificates and restore the publication point.

An adversary that controls the CA for a publication point can publish signed products that create more subtle types of DoS attacks against RPs. For example, such an attacker could create subordinate CA certificates with Subject Information Access (SIA) pointers that lead RPs on a "wild goose chase" looking for additional publication points and signed products. An attacker could publish certificates with very brief validity intervals, or CRLs and manifests that become "stale" very quickly. This sort of attack would cause RPs to access repositories more frequently, and that might interfere with legitimate accesses by other RPs.

An attacker with this capability could create very large numbers of ROAs to be processed (with prefixes that are consistent with the allocation for the CA), and correspondingly large manifests. An attacker could create very deep subtrees with many ROAs per publication point, etc. All of these types of DoS attacks against RPs are feasible within the syntactic and semantic constraints established for RPKI certificates, CRLs, and signed objects.

An attack that results in revocation and replacement (e.g., key rollover or certificate renewal) of a CA certificate would cause RPs to replace the old, valid certificate with the new one. This new certificate might contain a public key that does not correspond to the private key held by the certificate subject. That would cause objects signed by that subject to be rejected as invalid, and prevent the affected subject from being able to sign new objects. As above, RPs would not consider as valid any ROAs issued under the affected CA certificate, and updates based on router certificates issued by the affected CA would be rejected. This would make routes dependent on these signed products to be less preferred. However, the constraints imposed by the use of RFC 3779 [RFC3779] extensions do prevent a compromised CA from issuing (valid) certificates with INRs outside the scope of the CA, thus limiting the impact of the attack.

An adversary that controls a CA could issue CA certificates with overlapping INRs to different entities, when no transfer of INRs is intended. This could cause confusion for RPs as conflicting ROAs could be issued by the distinct (subordinate) CAs.

An adversary could replace a CA certificate, use the corresponding private key to issue new signed products, and then publish them at a publication point controlled by the attacker. This would effectively transfer the affected INRs to the adversary, or to a third party of his choosing. The result would be to cause RPs to view the entity that controls the private key in question as the legitimate INR holder. Again the constraints imposed by the use of RFC 3779 extensions prevent a compromised CA from issuing (valid) certificates with INRs outside the scope of the CA, thus limiting the impact of the attack.

Finally, an entity that manages a repository publication point can inadvertently act as an attacker (as first noted by Pogo). For example, a CA might fail to replace its own certificate in a timely fashion (well before it expires). If might fail to issue its CRL and manifest prior to expiration, creating stale instances of these products that cause concern for RPs. A CA with many subordinate CAs (e.g., an RIR or NIR) might fail to distribute the expiration times for the CA certificates that it issues. A network with many ROAs might do the same for the EE certificates associated with the ROAs it generates. A CA could rollover its key, but fail to reissue subordinate CA certificates under its new key. Poor planning with regard to rekey intervals for managed CAs could impose undue burdens for RPs, despite a lack of malicious intent. All of these example of mismanagement could adversely affect RPs, despite the absence of malicious intent.

5. Residual Vulnerabilities

The RPKI, upon which BGPSEC relies, has several residual vulnerabilities that were discussed in the preceding text (Section 4.4 and Section 4.5). These vulnerabilities are of two principle forms:

- the RPKI repository system may be attacked in ways that make its contents unavailable, not current, or inconsistent. The principle defense against most forms of DoS attacks is the use of a local cache by each RP. The local cache ensures availability of previously-acquired RPKI data, in the event that a repository is inaccessible or if repository contents are deleted (maliciously). Nonetheless, the system cannot ensure that every RP will always have access to up-to-date RPKI data. An RP, when it detects a problem with acquired repository data has two options:

  1. The RP may choose to make use of its local cache, employing local configuration settings that tolerate expired or stale objects. (Such behavior is, nominally, always within the purview of an RP in PKI.) Using cached, expired or stale data subjects the RP to attacks that take advantage of the RP's ignorance of changes to this data.

  2. The RP may chose to purge expired objects. Purging expired objects removes the security info associated with the real world INRs to which the objects refer. This is equivalent to the affected INRs not having been afforded protection via the RPKI. Since use of the RPKI (and BGPSEC) is voluntary, there may always be set of INRs that are not protected by these mechanisms. Thus purging moves the affected INRs to the set of non-participating INR holders. This more conservative response enables an attacker to move INRs from the protected to the unprotected set.

- any CA in the RPKI may misbehave within the bounds of the INRs allocated to it, e.g., it may issue certificates with duplicate resource allocations or revoke certificates inappropriately. This vulnerability is intrinsic in any PKI, but its impact is limited in the RPKI because of the use or RFC 3779 extensions. It is anticipated that RPs will deal with such misbehavior through administrative means, once it is detected.

BGPSEC has a separate set of residual vulnerabilities:

- "Route leaks" are viewed as a routing security problem by many network operators, even though there is no IETF-codified definition of a route leak. BGP itself does not include semantics that preclude what many perceive as route leaks. Moreover, route leaks are outside the scope of BGPSEC, at this time, based on the SIDR charter. Thus route leaks are not addressed in this threat model.

- BGPSEC signatures do not protect all attributes associated with an AS_path. Some of these attributes are employed as inputs to routing decisions. Thus attacks that modify (or strip) these other attributes are not detected by BGPSEC. The SIDR charter calls for protecting only the info needed to verify that a received route traversed the ASes on question, and that the NLRI in the route is what was advertised. Thus, protection of other attributes is outside the scope of the charter, at the time this document was prepared.

- BGPSEC cannot ensure that an AS will withdraw a route when the AS no longer has a route for a prefix, as noted in Section 4.2. BGPSEC may incorporate features to limit the lifetime of an advertisement. Such lifetime limits provide an upper bound on the time that the failure to withdraw a route will remain effective.

6. Security Considerations

A threat model is, by definition, a security-centric document. Unlike a protocol description, a threat model does not create security problems nor purport to address security problems. This model postulates a set of threats (i.e., motivated, capable adversaries) and examines classes of attacks that these threats are capable of effecting, based on the motivations ascribed to the threats. It describes the impact of these types of attacks on BGPSEC, including on the RPKI on which BGPSEC relies. It describes how the design of the RPKI (and the current BGPSEC design) address classes of attacks, where applicable. It also notes residual vulnerabilities.

7. IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this version of the document.]

8. Acknowledgements

The author wishes to thank...

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[Kent2000] Kent, S., Lynn, C., and K. Seo, "Design and Analysis of the Secure Border Gateway Protocol (S-BGP)", IEEE DISCEX Conference, June 2000.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, February 2012.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, February 2012.

[TCPMD5] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

Authors' Addresses

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
US

Email: kent@bbn.com

Andrew Chi
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
US

Email: achi@bbn.com

Secure Inter-Domain Routing
Internet-Draft
Intended status: Informational
Expires: June 14, 2014

S. Kent
BBN
A. Chi
UNC-CH

December 11, 2013

Threat Model for BGP Path Security
draft-ietf-sidr-bgpsec-threats-09

Abstract

This document describes a threat model for the context in which Exterior Border Gateway Protocol (EBGP) path security mechanisms will be developed. The threat model includes an analysis of the Resource Public Key Infrastructure (RPKI), and focuses on the ability of an autonomous system (AS) to verify the authenticity of the AS path info received in a BGP update. We use the term PATHSEC to refer to any BGP path security technology that makes use of the RPKI. PATHSEC will secure BGP, consistent with the inter-AS security focus of the RPKI.

The document characterizes classes of potential adversaries that are considered to be threats, and examines classes of attacks that might be launched against PATHSEC. It does not revisit attacks against unprotected BGP, as that topic has already been addressed in the BGP-4 standard. It concludes with brief discussion of residual vulnerabilities.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the security context in which PATHSEC is intended to operate. The term "PATHSEC" (for path security) refers to any design used to preserve the integrity and authenticity of the AS_PATH attribute carried in a BGP update message [RFC4271]. The security context used throughout this document is established by the SIDR charter [SIDR-CH]. The charter requires that solutions that afford PATHSEC make use of the Resource Public Key Infrastructure (RPKI) [RFC6480]. It also calls for protecting only the information required to verify that a received route traversed the Autonomous Systems (ASes) in question, and that the Network Layer Reachability Information (NLRI) in the route is what was advertised.

Thus the goal of PATHSEC is to enable a BGP speaker to verify that the ASes enumerated in this path attribute represent the sequence of

ASes that the NLRI traversed. The term PATHSEC is thus consistent with the goal described above. (Other SIDR documents use the term "BGPSEC" to refer to a specific design, thus we avoid use of that term here.)

This document discusses classes of potential adversaries that are considered to be threats, and classes of attacks that might be launched against PATHSEC. Because PATHSEC will rely on the RPKI, threats and attacks against the RPKI are included. This model also takes into consideration classes of attacks that are enabled by the use of PATHSEC (e.g., based on use of the RPKI).

The motivation for developing PATHSEC, i.e., residual security concerns for BGP, is well described in several documents, including "BGP Security Vulnerabilities Analysis" [RFC4272] and "Design and Analysis of the Secure Border Gateway Protocol (S-BGP)" [Kent2000]. All of these documents note that BGP does not include mechanisms that allow an Autonomous System (AS) to verify the legitimacy and authenticity of BGP route advertisements. (BGP now mandates support for mechanisms to secure peer-peer communication, i.e., for the links that connect BGP routers. There are several secure protocol options to addresses this security concern, e.g., IPsec [RFC4301] and TCP-AO [RFC5925]. This document briefly notes the need to address this aspect of BGP security, but focuses on application layer BGP security issues that must be addressed by PATHSEC.)

RFC 4272 [RFC4272] succinctly notes:

> "BGP speakers themselves can inject bogus routing information, either by masquerading as any other legitimate BGP speaker, or by distributing unauthorized routing information as themselves. Historically, misconfigured and faulty routers have been responsible for widespread disruptions in the Internet. The legitimate BGP peers have the context and information to produce believable, yet bogus, routing information, and therefore have the opportunity to cause great damage. The cryptographic protections of [TCPMD5] and operational protections cannot exclude the bogus information arising from a legitimate peer. The risk of disruptions caused by legitimate BGP speakers is real and cannot be ignored."

PATHSEC is intended to address the concerns cited above, to provide significantly improved path security, building upon the route origination validation capability offered by use of the RPKI [RFC6810]. Specifically, the RPKI enables relying parties (RPs) to determine if the origin AS for a path was authorized to advertise the prefix contained in a BGP update message. This security feature is enabled by the use of two types of digitally signed data: a PKI

[RFC6487] that associates one or more prefixes with the public key(s) of an address space holder, and Route Origination Authorizations (ROAs) [RFC6482] that allows a prefix holder to specify the AS(es) that are authorized to originate routes for a prefix.

The security model adopted for PATHSEC does not assume an "oracle" that can see all of the BGP inputs and outputs associated with every AS or every BGP router. Instead, the model is based on a local notion of what constitutes legitimate, authorized behavior by the BGP routers associated with an AS. This is an AS-centric model of secure operation, consistent with the AS-centric model that BGP employs for routing. This model forms the basis for the discussion that follows.

This document begins with a brief set of definitions relevant to the subsequent sections. It then discusses classes of adversaries that are perceived as viable threats against routing in the public Internet. It continues to explore a range of attacks that might be effected by these adversaries, against both path security and the infrastructure upon which PATHSEC relies. It concludes with a brief review of residual vulnerabilities, i.e., vulnerabilities that are not addressed by use of the RPKI and that appear likely to be outside the scope of PATHSEC mechanisms.

2. Terminology

The following security and routing terminology definitions are employed in this document.

Adversary - An adversary is an entity (e.g., a person or an organization) perceived as malicious, relative to the security policy of a system. The decision to characterize an entity as an adversary is made by those responsible for the security of a system. Often one describes classes of adversaries with similar capabilities or motivations, rather than specific individuals or organizations.

Attack - An attack is an action that attempts to violate the security policy of a system, e.g., by exploiting a vulnerability. There is often a many to one mapping of attacks to vulnerabilities, because many different attacks may be used to exploit a vulnerability.

Autonomous System (AS) - An AS is a set of one or more IP networks operated by a single administrative entity.

AS Number (ASN) - An ASN is a 2 or 4 byte number issued by a registry to identify an AS in BGP.

Certification Authority (CA) - An entity that issues digital certificates (e.g., X.509 certificates) and vouches for the binding between the data items in a certificate.

Countermeasure - A countermeasure is a procedure or technique that thwarts an attack, preventing it from being successful. Often countermeasures are specific to attacks or classes of attacks.

Border Gateway Protocol (BGP) - A path vector protocol used to convey "reachability" information among autonomous systems, in support of inter-domain routing.

False (Route) Origination - If a network operator originates a route for a prefix that the operator does not hold (and that it has not been authorized to originate by the prefix holder, this is termed false route origination.

Internet Service Provider (ISP) - An organization managing (and, typically, selling,) Internet services to other organizations or individuals.

Internet Number Resources (INRs) - IPv4 or IPv6 address space and ASNs

Internet Registry - An organization that manages the allocation or distribution of INRs. This encompasses the Internet Assigned Number Authority (IANA), Regional Internet Registries (RIRs), National Internet Registries (NIRs), and Local Internet Registries (LIRs, network operators).

Man in the Middle (MITM) - A MITM is an entity that is able to examine and modify traffic between two (or more) parties on a communication path.

Network Operator - An entity that manages an AS and thus emits (E)BGP updates, e.g., an ISP.

NOC (Network Operations Center) - A network operator employs a set equipment and a staff to manage a network, typically on a 24/7 basis. The equipment and staff are often referred to as the NOC for the network.

Prefix - A prefix is an IP address and a mask used to specify a set of addresses that are grouped together for purposes of routing.

Public Key Infrastructure (PKI) - A PKI is a collection of hardware, software, people, policies, and procedures used to create, manage, distribute, store, and revoke digital certificates.

Relying Parties (RPs) - An RP is an entity that makes use of signed products from a PKI, i.e., relies on signed data that is verified using certificates and Certificate Revocation Lists (CRLs) from a PKI.

RPKI Repository System - The RPKI repository system consists of a distributed set of loosely synchronized databases.

Resource PKI (RPKI) - A PKI operated by the entities that manage INRs, and that issues X.509 certificates (and CRLs) that attest to the holdings of INRs.

RPKI Signed Object - An RPKI signed object is a Cryptographic Message Syntax (CMS)-encapsulated data object complying with the format and semantics defined in [RFC6488].

Route - In the Internet, a route is a prefix and an associated sequence of ASNs that indicates a path via which traffic destined for the prefix can be directed. (The route includes the origin AS.)

Route leak - A route leak is said to occur when AS-A advertises routes that it has received from an AS-B to AS-A’s neighbors, but AS-A is not viewed as a transit provider for the prefixes in the route.

Threat - A threat is a motivated, capable adversary. An adversary that is not motivated to launch an attack is not a threat. An adversary that is motivated but not capable of launching an attack also is not a threat.

Vulnerability - A vulnerability is a flaw or weakness in a system’s design, implementation, or operation and management that could be exploited to violate the security policy of a system.

3. Threat Characterization

As noted in Section 2 above, a threat is defined as a motivated, capable, adversary. The following classes of threats represent classes of adversaries viewed as relevant to this environment.

Network Operators - A network operator may be a threat. An operator may be motivated to cause BGP routers it controls to emit update messages with inaccurate routing info, e.g., to cause traffic to flow via paths that are economically advantageous for the operator. Such updates might cause traffic to flow via paths that would otherwise be rejected as less advantageous by other network operators. Because an operator controls the BGP routers in its network, it is in a position to modify their operation in arbitrary ways. Routers managed by a network operator are vehicles for mounting MITM attacks on both control and data plane traffic. If an operator participates in the RPKI, it will have at least one CA resource certificate and may be able to generate an arbitrary number of subordinate CA certificates and ROAs. It will be authorized to populate (and may even host) its own repository publication point. If it implements PATHSEC, and if PATHSEC makes use of certificates associated with routers or ASes, it will have the ability to issue such certificates for itself. If PATHSEC digitally signs updates, it will be able to do so in a fashion that will be accepted by PATHSEC-enabled neighbors.

Hackers - Hackers are considered a threat. A hacker might assume control of network management computers and routers controlled by operators, including operators that implement PATHSEC. In such cases, hackers would be able to act as rogue network operators (see above). It is assumed that hackers generally do not have the capability to effect MITM attacks on most links between networks (links used to transmit BGP and subscriber traffic). A hacker might be recruited, without his/her knowledge, by criminals or by nations, to act on their behalf. Hackers may be motivated by a desire for "bragging rights" or for profit or to express support for a cause ("hacktivists" [Sam04]). We view hackers as possibly distinct from criminals in that the former are presumed to effect attacks only remotely (not via a physical presence associated with a target) and not necessarily for monetary gain. Some hackers may commit criminal acts (depending on the jurisdiction), and thus there is a potential for overlap between this adversary group and criminals.

Criminals - Criminals may be a threat. Criminals might persuade (via threats or extortion) a network operator to act as a rogue operator (see above), and thus be able to effect a wide range of attacks. Criminals might persuade the staff of a telecommunications provider to enable MITM attacks on links between routers. Motivations for criminals may include the ability to extort money from network operators or network operator clients, e.g., by adversely affecting routing for these network operators or their clients. Criminals also may wish to manipulate routing to conceal the sources of spam, DoS attacks, or other criminal activities.

Registries - Any registry in the RPKI could be a threat. Staff at the registry are capable of manipulating repository content or mismanaging the RPKI certificates that they issue. These actions could adversely affect a network operator or a client of a network operator. The staff could be motivated to do this based on political pressure from the nation in which the registry operates (see below) or due to criminal influence (see above).

Nations - A nation may be a threat. A nation may control one or more network operators that operate in the nation, and thus can cause them to act as rogue network operators. A nation may have a technical active wiretapping capability (e.g., within its territory) that enables it to effect MITM attacks on inter-network traffic. (This capability may be facilitated by control or influence over a telecommunications provider operating within the nation.) It may have an ability to attack and take control of routers or management network computers of network operators in other countries. A nation may control a registry (e.g., an RIR) that operates within its territory, and might force that registry to act in a rogue capacity. National threat motivations include the desire to control the flow of traffic to/from the nation or to divert traffic destined for other nations (for passive or active wiretapping, including DoS).

4. Attack Characterization

This section describes classes of attacks that may be effected against Internet routing (relative to the context described in Section 1). Attacks are classified based on the target of the attack, as an element of the routing system, or the routing security infrastructure on which PATHSEC relies. In general, attacks of interest are ones that attempt to violate the integrity or authenticity of BGP traffic, or which violate the authorizations associated with entities participating in the RPKI. Attacks that violate the implied confidentiality of routing traffic, e.g., passive wiretapping attacks, are not considered a requirement for BGP security (see [RFC4272]).

4.1. Active wiretapping of sessions between routers

An adversary may attack the BGP (TCP) session that connects a pair of BGP speakers. An active attack against a BGP (TCP) session can be effected by directing traffic to a BGP speaker from some remote point, or by being positioned as a MITM on the link that carries BGP session traffic. Remote attacks can be effected by any adversary. A MITM attack requires access to the link. Modern transport networks may be as complex as the packet networks that utilize them for inter-AS links. Thus these transport networks may present significant attack surfaces. Nonetheless, only some classes of adversaries are assumed to be capable of MITM attacks against a BGP session. MITM attacks may be directed against BGP, PATHSEC-protected BGP, or against TCP or IP. Such attacks include replay of selected BGP messages, selective modification of BGP messages, and DoS attacks against BGP routers. [RFC4272] describes several countermeasures for such attacks, and thus this document does not further address such attacks.

4.2. Attacks on a BGP router

An adversary may attack a BGP router, whether it implements PATHSEC or not. Any adversary that controls routers legitimately, or that can assume control of a router, is assumed to be able to effect the types of attacks described below. Note that any router behavior that can be ascribed to a local routing policy decision is not considered to be an attack. This is because such behavior could be explained as a result of local policy settings, and thus is beyond the scope of what PATHSEC can detect as unauthorized behavior. Thus, for example, a router may fail to propagate some or all route withdrawals or effect "route leaks". (These behaviors are not precluded by the specification for BGP, and might be the result of a local policy that is not publicly disclosed. As a result, they are not considered attacks. See Section 5 for additional discussion.)

Attacks on a router are equivalent to active wiretapping attacks (in the most general sense) that manipulate (forge, tamper with, or suppress) data contained in BGP updates. The list below illustrates attacks of this type.

AS Insertion: A router might insert one or more ASNs, other than its own ASN, into an update message. This violates the BGP spec and thus is considered an attack.

False (Route) Origination: A router might originate a route for a prefix, when the AS that the router represents is not authorized to originate routes for that prefix. This is an attack, but it is addressed by the use of the RPKI [RFC6480].

Secure Path Downgrade: A router might remove AS_PATH data from a PATHSEC-protected update that it receives, when forwarding this update to a PATHSEC-enabled neighbor. This behavior violates the PATHSEC security goals and thus is considered an attack.

Invalid AS_PATH Data Insertion: A router might emit a PATHSEC-protected update with "bad" data (such as a signature), i.e., PATHSEC data that cannot be validated by other PATHSEC routers. Such behavior is assumed to violate the PATHSEC goals and thus is considered an attack.

Stale Path Announcement: If PATHSEC-secured announcements can expire, such an announcement may be propagated with PATHSEC data that is "expired". This behavior would violate the PATHSEC goals and is considered a type of replay attack.

Premature Path Announcement Expiration: If a PATHSEC-secured announcement has an associated expiration time, a router might emit a PATHSEC-secured announcement with an expiry time that is very short. Unless the PATHSEC protocol specification mandates a minimum expiry time, this is not an attack. However, if such a time is mandated, this behavior becomes an attack. BGP speakers along a path generally cannot determine if an expiry time is "suspiciously short" since they cannot know how long a route may have been held by an earlier AS, prior to being released.

MITM Attack: A cryptographic key used for point-to-point security (e.g., TCP-AO, TLS, or IPsec) between two BGP routers might be compromised (e.g., by extraction from a router). This would enable an adversary to effect MITM attacks on the link(s) where the key is used. Use of specific security mechanisms to protect inter-router links between ASes is outside the scope of PATHSEC.

Compromised Router Private Key: If PATHSEC mechanisms employ public key cryptography, e.g., to digitally sign data in an update, then a private key associated with a router or an AS might be compromised by an attack against the router. An adversary with access to this key would be able to generate updates that appear to have passed through the AS that this router represents. Such updates might be in injected on a link between the compromised router and its neighbors, if that link is accessible to the adversary. If the adversary controls another network, it could use this key to forge signatures that appear to come from the AS or router(s) in question, with some constraints. So, for example, an adversary that controls another AS could use a compromised router/AS key to issue PATHSEC-signed data that include the targeted AS/router. (Neighbors of the adversary’s AS ought not accept a route that purports to emanate directly from the targeted AS. So, an adversary could take a legitimate, protected route that passes through the compromised AS, add itself as the next hop, and then forward the resulting route to neighbors.)

Withdrawal Suppression Attack: A PATHSEC-protected update may be signed and announced, and later withdrawn. An adversary controlling intermediate routers could fail to propagate the withdrawal. BGP is already vulnerable to behavior of this sort, so withdrawal suppression is not characterized as an attack, under the assumptions upon which this mode is based (i.e., no oracle).

4.3. Attacks on network operator management computers (non-CA computers)

An adversary may choose to attack computers used by a network operator to manage its network, especially its routers. Such attacks might be effected by an adversary who has compromised the security of these computers. This might be effected via remote attacks, extortion of network operations staff, etc. If an adversary compromises NOC computers, he can execute any management function that authorized network operations staff would have performed. Thus the adversary could modify local routing policy to change preferences, to black-hole certain routes, etc. This type of behavior cannot be externally detected as an attack. Externally, this appears as a form of rogue operator behavior. (Such behavior might be perceived as accidental or malicious by other operators.)

If a network operator participates in the RPKI, an adversary could manipulate the RP tools that extract data from the RPKI, causing the output of these tools to be corrupted in various ways. For example, an attack of this sort could cause the operator to view valid routes as not validated, which could alter its routing behavior.

If an adversary invoked the tool used to manage the repository publication point for this operator, it could delete any objects stored there (certificates, CRLs, manifests, ROAs, or subordinate CA certificates). This could affect the routing status of entities that have allocations/assignments from this network operator (e.g., by deleting their CA certificates).

An adversary could invoke the tool used to request certificate revocation, causing router certificates, ROAs, or subordinate CA certificates to be revoked. An attack of this sort could affect not only this operator, but also any operators that receive allocations/assignments from it, e.g., because their CA certificates were revoked.

If an operator is PATHSEC-enabled, an attack of this sort could cause the affected operator to be viewed as not PATHSEC-enabled, possibly making routes it emits be less preferred by other operators.

If an adversary invoked a tool used to request ROAs, it could effectively re-allocate some of the prefixes allocated/assigned to the network operator (e.g., by modifying the origin AS in ROAs). This might cause other PATHSEC-enabled networks to view the affected network as no longer originating routes for these prefixes. Multi-homed subscribers of this operator who received an allocation from the operator might find their traffic was now routed via other connections.

If the network operator is PATHSEC-enabled, and make use of certificates associated with routers/ASes, an adversary could invoke a tool used to request such certificates. The adversary could then replace valid certificates for routers/ASes with ones that might be rejected by PATHSEC-enabled neighbors.

4.4. Attacks on a repository publication point

A critical element of the RPKI is the repository system. An adversary might attack a repository, or a publication point within a repository, to adversely affect routing.

This section considers only those attacks that can be launched by any adversary who controls a computer hosting one or more repository publication points, without access to the cryptographic keys needed to generate valid RPKI signed products. Such attacks might be effected by an insider or an external threat. Because all repository objects are digitally signed, attacks of this sort translate into DoS attacks against the RPKI RPs. There are a few distinct forms of such attacks, as described below.

Note first that the RPKI calls for RPs to cache the data they acquire and verify from the repository system [RFC6480][RFC6481]. Attacks that delete signed products, that insert products with "bad" signatures, that tamper with object signatures, or that replace newer objects with older (valid) ones, can be detected by RPs (with a few exceptions). RPs are expected to make use of local caches. If repository publication points are unavailable or the retrieved data is corrupted, an RP can revert to using the cached data. This behavior helps insulate RPs from the immediate effects of DoS attacks on publication points.

Each RPKI data object has an associated date at which it expires, or is considered stale. (Certificates expire, CRLs become stale.) When an RP uses cached data it is a local decision how to deal with stale or expired data. It is common in PKIs to make use of stale certificate revocation status data, when fresher data is not available. Use of expired certificates is less common, although not unknown. Each RP will decide, locally, whether to continue to make use of or ignore cached RPKI objects that are stale or expired.

If an adversary inserts an object into a publication point, and the object has a "bad" signature, the object will not be accepted and used by RPs.

If an adversary modifies any signed product at a publication point, the signature on the product will fail, causing RPs to not accept it. This is equivalent to deleting the object, in many respects.

If an adversary deletes one or more CA certificates, ROAs or the CRL for a publication point, the manifest for that publication point will allow an RP to detect this attack. An RP can continue to use the last valid instance of the deleted object (as a local policy option), thus minimizing the impact of such an attack.

If an adversary deletes a manifest (and does not replace it with an older instance), that is detectable by RPs. Such behavior should result in the CA (or publication point maintainer) being notified of the problem. An RP can continue to use the last valid instance of the deleted manifest (a local policy option), thus minimizing the impact of such an attack.

If an adversary deletes newly added CA certificates or ROAs, and replaces the current manifest with the previous manifest, the manifest (and the CRL that it matches) will be "stale" (see [RFC6486]). This alerts an RP that there may be a problem. The RP should use the information from a Ghostbuster record [RFC6493] to contact the entity responsible for the publication point, requesting that entity to remedy the problem (e.g., republish the missing CA certificates and/or ROAs). An RP cannot know the content of the new certificates or ROAs that are not present, but it can continue to use what it has cached. An attack of this sort will, at least temporarily, cause RPs to be unaware of the newly published objects. INRs associated with these objects will be treated as unauthenticated.

If a CA revokes a CA certificate or a ROA (via deleting the corresponding EE certificate), and the adversary tries to reinstate that CA certificate or ROA, the adversary would have to rollback the CRL and the manifest to undo this action by the CA. As above, this would make the CRL and manifest stale, and this is detectable by RPs. An RP cannot know which CA certificates or ROAs were deleted. Depending on local policy, the RP might use the cached instances of the affected objects, and thus be tricked into making decisions based on these revoked objects. Here too the goal is that the CA will be notified of the problem (by RPs) and will remedy the error.

In the attack scenarios above, when a CRL or manifest is described as stale, this means that the next issue date for the CRL or manifest has passed. Until the next issue date, an RP will not detect the attack. Thus it behooves CAs to select CRL/manifest lifetimes (the two are linked) that represent an acceptable trade-off between risk and operational burdens.

Attacks effected by adversaries that are legitimate managers of publication points can have much greater effects, and are discussed below under attacks on or by CAs.

4.5. Attacks on an RPKI CA

Every entity to which INRs have been allocated/assigned is a CA in the RPKI. Each CA is nominally responsible for managing the repository publication point for the set of signed products that it generates. (An INR holder may choose to outsource the operation of the RPKI CA function, and the associated publication point. In such cases, the organization operating on behalf of the INR holder becomes the CA, from an operational and security perspective. The following discussion does not distinguish such outsourced CA operations.)

Note that attacks attributable to a CA may be the result of malice by the CA (i.e., the CA is the adversary) or they may result from a compromise of the CA.

All of adversaries listed in Section 2 are presumed to be capable of launching attacks against the computers used to perform CA functions. Some adversaries might effect an attack on a CA by violating personnel or physical security controls as well. The distinction between CA as adversary vs. CA as an attack victim is important. Only in the latter case should one expect the CA to remedy problems caused by a attack once the attack has been detected. (If a CA does not take such action, the effects are the same as if the CA is an adversary.)

Note that most of the attacks described below do not require disclosure of a CA’s private key to an adversary. If the adversary can gain control of the computer used to issue certificates, it can effect these attacks, even though the private key for the CA remains "secure" (i.e., not disclosed to unauthorized parties). However, if the CA is not the adversary, and if the CA’s private key is not compromised, then recovery from these attacks is much easier. This motivates use of hardware security modules to protect CA keys, at least for higher tiers in the RPKI.

An attack by a CA can result in revocation or replacement of any of the certificates that the CA has issued. Revocation of a certificate should cause RPs to delete the (formerly) valid certificate (and associated signed object, in the case of a revoked EE certificate) that they have cached. This would cause repository objects (e.g., CA certificates and ROAs) that are verified under that certificate to be considered invalid, transitively. As a result, RPs would not consider as valid any ROAs or PATHSEC-protected updates based on these certificates, which would make routes dependent on them to be less preferred. Because a CA that revokes a certificate is authorized to do so, this sort of attack cannot be detected, intrinsically, by most RPs. However, the entities affected by the revocation or replacement of CA certificates can be expected to detect the attack and contact the CA to effect remediation. If the CA was not the adversary, it should be able to issue new certificates and restore the publication point.

An adversary that controls the CA for a publication point can publish signed products that create more subtle types of DoS attacks against RPs. For example, such an attacker could create subordinate CA certificates with Subject Information Access (SIA) pointers that lead RPs on a "wild goose chase" looking for additional publication points and signed products. An attacker could publish certificates with very brief validity intervals, or CRLs and manifests that become "stale" very quickly. This sort of attack would cause RPs to access repositories more frequently, and that might interfere with legitimate accesses by other RPs.

An attacker with this capability could create very large numbers of ROAs to be processed (with prefixes that are consistent with the allocation for the CA), and correspondingly large manifests. An attacker could create very deep subtrees with many ROAs per publication point, etc. All of these types of DoS attacks against RPs are feasible within the syntactic and semantic constraints established for RPKI certificates, CRLs, and signed objects.

An attack that results in revocation and replacement (e.g., key rollover or certificate renewal) of a CA certificate would cause RPs to replace the old, valid certificate with the new one. This new certificate might contain a public key that does not correspond to the private key held by the certificate subject. That would cause objects signed by that subject to be rejected as invalid, and prevent the affected subject from being able to sign new objects. As above, RPs would not consider as valid any ROAs issued under the affected CA certificate, and updates based on router certificates issued by the affected CA would be rejected. This would make routes dependent on these signed products to be less preferred. However, the constraints imposed by the use of RFC 3779 [RFC3779] extensions do prevent a compromised CA from issuing (valid) certificates with INRs outside the scope of the CA, thus limiting the impact of the attack.

An adversary that controls a CA could issue CA certificates with overlapping INRs to different entities, when no transfer of INRs is intended. This could cause confusion for RPs as conflicting ROAs could be issued by the distinct (subordinate) CAs.

An adversary could replace a CA certificate, use the corresponding private key to issue new signed products, and then publish them at a publication point controlled by the attacker. This would effectively transfer the affected INRs to the adversary, or to a third party of his choosing. The result would be to cause RPs to view the entity that controls the private key in question as the legitimate INR holder. Again the constraints imposed by the use of RFC 3779 extensions prevent a compromised CA from issuing (valid) certificates with INRs outside the scope of the CA, thus limiting the impact of the attack.

Finally, an entity that manages a repository publication point can inadvertently act as an attacker (an example of Walt Kelly's most famous "Pogo" quote [Kelly70]). For example, a CA might fail to replace its own certificate in a timely fashion (well before it expires). If might fail to issue its CRL and manifest prior to expiration, creating stale instances of these products that cause concern for RPs. A CA with many subordinate CAs (e.g., an RIR or NIR) might fail to distribute the expiration times for the CA certificates that it issues. A network with many ROAs might do the same for the EE certificates associated with the ROAs it generates. A CA could rollover its key, but fail to reissue subordinate CA certificates under its new key. Poor planning with regard to rekey intervals for managed CAs could impose undue burdens for RPs, despite a lack of malicious intent. All of these example of mismanagement could adversely affect RPs, despite the absence of malicious intent.

5. Residual Vulnerabilities

The RPKI, upon which PATHSEC relies, has several residual vulnerabilities that were discussed in the preceding text (Section 4.4 and Section 4.5). These vulnerabilities are of two principle forms:

- the RPKI repository system may be attacked in ways that make its contents unavailable, not current, or inconsistent. The principle defense against most forms of DoS attacks is the use of a local cache by each RP. The local cache ensures availability of previously-acquired RPKI data, in the event that a repository is inaccessible or if repository contents are deleted (maliciously). Nonetheless, the system cannot ensure that every RP will always have access to up-to-date RPKI data. An RP, when it detects a problem with acquired repository data has two options:

  1. The RP may choose to make use of its local cache, employing local configuration settings that tolerate expired or stale objects. (Such behavior is, nominally, always within the purview of an RP in PKI.) Using cached, expired or stale data subjects the RP to attacks that take advantage of the RP's ignorance of changes to this data.

  2. The RP may chose to purge expired objects. Purging expired objects removes the security info associated with the real

world INRs to which the objects refer. This is equivalent to the affected INRs not having been afforded protection via the RPKI. Since use of the RPKI (and PATHSEC) is voluntary, there may always be set of INRs that are not protected by these mechanisms. Thus purging moves the affected INRs to the set of non-participating INR holders. This more conservative response enables an attacker to move INRs from the protected to the unprotected set.

- any CA in the RPKI may misbehave within the bounds of the INRs allocated to it, e.g., it may issue certificates with duplicate resource allocations or revoke certificates inappropriately. This vulnerability is intrinsic in any PKI, but its impact is limited in the RPKI because of the use of RFC 3779 extensions. It is anticipated that RPs will deal with such misbehavior through administrative means, once it is detected.

PATHSEC has a separate set of residual vulnerabilities:

- It has been stated that "route leaks" are viewed as a routing security problem by many operators. However, BGP itself does not include semantics that preclude what many perceive as route leaks, and there is no definition of the term in any RFC. This makes it inappropriate to address route leaks in this document. Additionally, route leaks are outside the scope of PATHSEC, consistent with the security context noted in Section 1 of this document. If, at a later time, the SIDR security context is revised to include route leaks, and an appropriate definition exists, this document should be revised.

- PATHSEC is not required to protect all attributes associated with an AS_PATH, even though some of these attributes may be employed as inputs to routing decisions. Thus attacks that modify (or strip) these other attributes are not prevented/detected by PATHSEC. As noted in Section 1, the SIDR security context calls for protecting only the info needed to verify that a received route traversed the ASes in question, and that the NLRI in the route is what was advertised. (The AS_PATH data also may have traversed ASes within a confederation that are not represented. However, these ASes are not externally visible, and thus do not influence route selection, so their omission in this context is not a security concern.) Thus, protection of other attributes is outside the scope of this document, as described in Section 1. If, at a later time, the SIDR security context is revised to include protection of additional BGP attributes, this document should be revised.

- PATHSEC cannot ensure that an AS will withdraw a route when the AS no longer has a route for a prefix, as noted in Section 4.2. PATHSEC may incorporate features to limit the lifetime of an advertisement. Such lifetime limits provide an upper bound on the time that the failure to withdraw a route will remain effective.

6. Security Considerations

A threat model is, by definition, a security-centric document. Unlike a protocol description, a threat model does not create security problems nor purport to address security problems. This model postulates a set of threats (i.e., motivated, capable adversaries) and examines classes of attacks that these threats are capable of effecting, based on the motivations ascribed to the threats. It describes the impact of these types of attacks on PATHSEC, including on the RPKI on which PATHSEC relies. It describes how the design of the RPKI (and the PATHSEC design goals) address classes of attacks, where applicable. It also notes residual vulnerabilities.

7. IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this version of the document.]

8. Acknowledgements

TBD

9. Informative References

[Kelly70] Kelly, W., "'We Have Met the Enemy, and He is Us': Pogo Earth Day Poster", April 1970.

[Kent2000] Kent, S., Lynn, C., and K. Seo, "Design and Analysis of the Secure Border Gateway Protocol (S-BGP)", IEEE DISCEX Conference, June 2000.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, February 2012.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, February 2012.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, February 2012.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, January 2013.

[SIDR-CH] "Secure Inter-Domain Routing: Charter for Working Group", September 2013, <http://tools.ietf.org/wg/sidr/charters?item=charter-sidr-2013-09-20.txt>.

[Sam04] Samuel, A., "Hacktivism and the Future of Political Participation", Ph.D. dissertation, Harvard University, August 2004.

Authors' Addresses

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
US

Email: kent@bbn.com

Andrew Chi
University of North Carolina - Chapel Hill
c/o Department of Computer Science
CB 3175, Sitterson Hall
Chapel Hill, NC 27599
US

Email: achi@cs.unc.edu

Network Working Group
Internet-Draft
Intended status: BCP
Expires: September 11, 2012

R. Bush
Internet Initiative Japan

March 10, 2012

RPKI-Based Origin Validation Operation
draft-ietf-sidr-origin-ops-15

Abstract

Deployment of RPKI-based BGP origin validation has many operational considerations. This document attempts to collect and present them. It is expected to evolve as RPKI-based origin validation is deployed and the dynamics are better understood.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

RPKI-based origin validation relies on widespread deployment of the Resource Public Key Infrastructure (RPKI) [RFC6480]. How the RPKI is distributed and maintained globally is a serious concern from many aspects.

The global RPKI is in very initial stages of deployment, there is no single root trust anchor, initial testing is being done by the IANA and the RIRs, and there are technical testbeds. It is thought that origin validation based on the RPKI will be deployed incrementally over the next year to five years. It is assumed that eventually there will be a single root trust anchor for the public address space.

Origin validation needs to be done only by an AS’s border routers and is designed so that it can be used to protect announcements which are originated by any network participating in Internet BGP routing: large providers, upstreams and down-streams, and by small stub/enterprise/edge routers.

Origin validation has been designed to be deployed on current routers without significant hardware upgrade. It should be used in border routers by operators from large backbones to small stub/entetprise/edge networks.

RPKI-based origin validation has been designed so that, with prudent local routing policies, there is little risk that what is seen as today’s normal Internet routing is threatened by imprudent deployment of the global RPKI, see Section 5.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], the RPKI, see [RFC6480], the RPKI Repository Structure, see [RFC6481], ROAs, see [RFC6482], the RPKI to Router Protocol, see [I-D.ietf-sidr-rpki-rtr], RPKI-based Prefix Validation, see [I-D.ietf-sidr-pfx-validate], and Ghostbusters Records, see [RFC6493].

3. RPKI Distribution and Maintenance

The RPKI is a distributed database containing certificates, CRLs, manifests, ROAs, and Ghostbusters Records as described in [RFC6481]. Policies and considerations for RPKI object generation and maintenance are discussed elsewhere.

A local relying party valid cache containing all RPKI data may be gathered from the global distributed database using the rsync protocol, [RFC5781], and a validation tool such as rcynic [rcynic].

Validated caches may also be created and maintained from other validated caches. Network operators SHOULD take maximum advantage of this feature to minimize load on the global distributed RPKI database. Of course, the recipient relying parties SHOULD re-validate the data.

Timing of inter-cache synchronization, and synchronization between caches and the global RPKI, is outside the scope of this document, and depends on things such as how often routers feed from the caches, how often the operator feels the global RPKI changes significantly, etc.

As inter-cache synchronization within an operator’s network does not impact global RPKI resources, an operator MAY choose to synchronize quite frequently.

As RPKI-based origin validation relies on the availability of RPKI data, operators SHOULD locate caches close to routers that require these data and services. ’Close’ is, of course, complex. One should consider trust boundaries, routing bootstrap reachability, latency, etc.

If insecure transports are used between an operator’s cache and their router(s), the Transport Security recommendations in [I-D.ietf-sidr-rpki-rtr] SHOULD be followed. In particular, operators MUST NOT use insecure transports between their routers and RPKI caches located in other Autonomous Systems.

For redundancy, a router SHOULD peer with more than one cache at the same time. Peering with two or more, at least one local and others remote, is recommended.

If an operator trusts upstreams to carry their traffic, they MAY also trust the RPKI data those upstreams cache, and SHOULD peer with caches made available to them by those upstreams. Note that this places an obligation on those upstreams to maintain fresh and reliable caches, and to make them available to their customers. And, as usual, the recipient SHOULD re-validate the data.

A transit provider or a network with peers SHOULD validate origins in announcements made by upstreams, down-streams, and peers. They still SHOULD trust the caches provided by their upstreams.

Before issuing a ROA for a super-block, an operator MUST ensure that all sub-allocations from that block which are announced by other ASs, e.g. customers, have correct ROAs in the RPKI. Otherwise, issuing a ROA for the super-block will cause the announcements of sub-allocations with no ROAs to be viewed as Invalid, see [I-D.ietf-sidr-pfx-validate].

Use of RPKI-based origin validation removes any need to originate more specifics into BGP to protect against mis-origination of a less specific prefix. Having a ROA for the covering prefix will protect it.

To aid translation of ROAs into efficient search algorithms in routers, ROAs SHOULD be as precise as possible, i.e. match prefixes as announced in BGP. E.g. software and operators SHOULD avoid use of excessive max length values in ROAs unless operationally necessary.

One advantage of minimal ROA length is that the forged origin attack does not work for sub-prefixes that are not covered by overly long max length. E.g. if, instead of 10.0.0.0/16-24, one issues 10.0.0.0/16 and 10.0.42.0/24, a forged origin attack can not succeed against 10.0.66.0/24. They must attack the whole /16, which is more likely to be noticed because of its size.

Therefore, ROA generation software MUST use the prefix length as the max length if the user does not specify a max length.

Operators SHOULD be conservative in use of max length in ROAs. E.g., if a prefix will have only a few sub-prefixes announced, multiple ROAs for the specific announcements SHOULD be used as opposed to one ROA with a long max length.

Operators owning prefix P should issue ROAs for all ASs which may announce P. If a prefix is legitimately announced by more than one AS, ROAs for all of the ASs SHOULD be issued so that all are considered Valid.

An environment where private address space is announced in eBGP the operator MAY have private RPKI objects which cover these private spaces. This will require a trust anchor created and owned by that environment, see [I-D.ietf-sidr-ltamgmt].

Operators issuing ROAs may have customers which announce their own prefixes and ASs into global eBGP but who do not wish to go though the work to manage the relevant certificates and ROAs. Operators SHOULD offer to provision the RPKI data for these customers just as they provision many other things for them.

While an operator using RPKI data MAY choose any polling frequency they wish for ensuring they have a fresh RPKI cache. However, if they use RPKI data as an input to operational routing decisions, they SHOULD ensure local caches inside their AS are synchronized with each other at least every four to six hours.

Operators should use tools which warn them of any impending ROA or certificate expiry which could affect the validity of their own data. Ghostbuster Records, see [RFC6493], can be used to facilitate contact with upstream CAs to effect repair.

4. Within a Network

Origin validation need only be done by edge routers in a network, those which border other networks/ASs.

A validating router will use the result of origin validation to influence local policy within its network, see Section 5. In deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy validation-capable border routers.

5. Routing Policy

Origin validation based on the RPKI marks a received announcement as having an origin which is Valid, NotFound, or Invalid, see [I-D.ietf-sidr-pfx-validate]. How this is used in routing SHOULD be specified by the operator's local policy.

Local policy using relative preference is suggested to manage the uncertainty associated with a system in early deployment, applying local policy to eliminate the threat of unreachability of prefixes due to ill-advised certification policies and/or incorrect certification data. E.g. until the community feels comfortable relying on RPKI data, routing on Invalid origin validity, though at a low preference, MAY occur.

As origin validation will be rolled out incrementally, coverage will be incomplete for a long time. Therefore, routing on NotFound validity state SHOULD be done for a long time. As the transition moves forward, the number of BGP announcements with validation state NotFound should decrease. Hence an operator's policy SHOULD NOT be overly strict, and should prefer Valid announcements, attaching a lower preference to, but still using, NotFound announcements, and dropping or giving a very low preference to Invalid announcements.

Some providers may choose to set Local-Preference based on the RPKI validation result. Other providers may not want the RPKI validation result to be more important than AS-path length -- these providers would need to map RPKI validation result to some BGP attribute that is evaluated in BGP’s path selection process after AS-path is evaluated. Routers implementing RPKI-based origin validation MUST provide such options to operators.

Local-Preference may be used to carry both the validity state of a prefix along with it’s traffic engineering characteristic(s). It is likely that an operator already using Local-Preference will have to change policy so they can encode these two separate characteristics in the same BGP attribute without negatively impact or opening privilege escalation attacks.

When using a metric which is also influenced by other local policy, an operator should be careful not to create privilege upgrade vulnerabilities. E.g. if Local Pref is set depending on validity state, be careful that peer community signaling MAY NOT upgrade an Invalid announcement to Valid or better.

Announcements with Valid origins SHOULD be preferred over those with NotFound or Invalid origins, if the latter are accepted at all.

Announcements with NotFound origins SHOULD be preferred over those with Invalid origins.

Announcements with Invalid origins SHOULD NOT be used, but MAY be used to meet special operational needs. In such circumstances, the announcement SHOULD have a lower preference than that given to Valid or NotFound.

Validity state signaling SHOULD NOT be accepted from a neighbor AS. The validity state of a received announcement has only local scope due to issues such as scope of trust, RPKI synchrony, and [I-D.ietf-sidr-ltamgmt].

6. Notes

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no ’fix’ for this, it is the nature of distributed data with distributed caches.

Operators should beware that RPKI caches are loosely synchronized, even within a single AS. Thus, changes to the validity state of prefixes could be different within an operator’s network. In addition, there is no guaranteed interval from when an RPKI cache is updated to when that new information may be pushed or pulled into a set of routers via this protocol. This may result in sudden shifts of traffic in the operator’s network, until all of the routers in the AS have reached equilibrium with the validity state of prefixes reflected in all of the RPKI caches.

It is hoped that testing and deployment will produce advice on relying party cache loading and timing.

There is some uncertainty about the origin AS of aggregates and what, if any, ROA can be used. The long range solution to this is the deprecation of AS-SETs, see [I-D.wkumari-deprecate-as-sets].

As reliable access to the global RPKI and an operator’s caches (and possibly other hosts, e.g. DNS root servers) is important, an operator SHOULD take advantage of relying party tools which report changes in BGP or RPKI data which would negatively affect validation of such prefixes.

Operators who manage certificates SHOULD associate RPKI Ghostbusters Records (see [RFC6493]) with each publication point they control. These are publication points holding the CRL, ROAs, and other signed objects issued by the operator, and made available to other ASs in support of routing on the public Internet.

As a router must evaluate certificates and ROAs which are time dependent, routers’ clocks MUST be correct to a tolerance of approximately an hour.

It is not reasonable to expect RPKI-based validation to run on routers which do not support Four-octet AS Numbers (see [RFC4893], as it is not reasonable to generate ROAs for AS 23456.

Servers should provide time service, such as [RFC5905], to client routers.

7. Security Considerations

As the BGP origin AS of an update is not signed, origin validation is open to malicious spoofing. Therefore, RPKI-based origin validation is expected to deal only with inadvertent mis-advertisement.

Origin validation does not address the problem of AS-Path validation. Therefore paths are open to manipulation, either malicious or accidental.

As BGP does not ensure that traffic will flow via the paths it advertises, the data plane may not follow the control plane.

Be aware of the class of privilege escalation issues discussed in Section 5 above.

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

The author wishes to thank Shane Amante, Rob Austein, Steve Bellovin, Jay Borkenhagen, Steve Kent, Pradosh Mohapatra, Chris Morrow, Sandy Murphy, Keyur Patel, Heather and Jason Schiller, John Scudder, Kotikalapudi Sriram, Maureen Stillman, and Dave Ward.

10. References

10.1. Normative References

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-04 (work in progress), December 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-03 (work in progress), October 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-26 (work in progress), February 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI

Scheme", RFC 5781, February 2010.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, February 2012.

10.2. Informative References

[I-D.wkumari-deprecate-as-sets]
Kumari, W., "Deprecation of BGP AS_SET, AS_CONFED_SET.", draft-wkumari-deprecate-as-sets-01 (work in progress), September 2010.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[rcynic] "rcynic read-me", <http://subvert-rpki.hactrn.net/rcynic/README>.

Author’s Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: May 25, 2014

R. Bush
Internet Initiative Japan

November 21, 2013

RPKI-Based Origin Validation Operation
draft-ietf-sidr-origin-ops-23

Abstract

Deployment of RPKI-based BGP origin validation has many operational considerations. This document attempts to collect and present those which are most critical. It is expected to evolve as RPKI-based origin validation continues to be deployed and the dynamics are better understood.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without normative meaning.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RPKI-based origin validation relies on widespread deployment of the Resource Public Key Infrastructure (RPKI) [RFC6480]. How the RPKI is distributed and maintained globally is a serious concern from many aspects.

While the global RPKI is in the early stages of deployment, there is no single root trust anchor, initial testing is being done by the RIRs, and there are technical testbeds. It is thought that origin validation based on the RPKI will continue to be deployed incrementally over the next few years. It is assumed that eventually there must be a single root trust anchor for the public address space, see [iab].

Origin validation needs to be done only by an AS’s border routers and is designed so that it can be used to protect announcements which are originated by any network participating in Internet BGP routing: large providers, upstreams and down-streams, and by small stub/enterprise/edge routers.

Origin validation has been designed to be deployed on current routers without significant hardware upgrade. It should be used in border routers by operators from large backbones to small stub/entetprise/edge networks.

RPKI-based origin validation has been designed so that, with prudent local routing policies, there is little risk that what is seen as today’s normal Internet routing is threatened by imprudent deployment of the global RPKI, see Section 5.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], the RPKI, see [RFC6480], the RPKI Repository Structure, see [RFC6481], Route Origin Authorizations (ROAs), see [RFC6482], the RPKI to Router Protocol, see [RFC6810], RPKI-based Prefix Validation, see [RFC6811], and Ghostbusters Records, see [RFC6493].

3. RPKI Distribution and Maintenance

The RPKI is a distributed database containing certificates, Certificate Revocation Lists (CRLs), manifests, ROAs, and Ghostbusters Records as described in [RFC6481]. Policies and considerations for RPKI object generation and maintenance are discussed elsewhere.

The RPKI repository design [RFC6481] anticipated a hierarchic organization of repositories, as this seriously improves the performance of relying parties gathering data over a non-hierarchic organization. Publishing parties MUST implement hierarchic directory structures.

A local relying party valid cache containing all RPKI data may be gathered from the global distributed database using the rsync protocol, [RFC5781], and a validation tool such as rcynic [rcynic].

A validated cache contains all RPKI objects that the RP has verified to be valid according to the rules for validation RPKI certificates and signed objects, see [RFC6487] and [RFC6488]. Entities that trust the cache can use these RPKI objects without further validation.

Validated caches may also be created and maintained from other validated caches. Network operators SHOULD take maximum advantage of this feature to minimize load on the global distributed RPKI database. Of course, the recipient relying parties should re-validate the data.

As Trust Anchor Locators (TALs), see [RFC6490], are critical to the RPKI trust model, operators should be very careful in their initial selection and vigilant in their maintenance.

Timing of inter-cache synchronization, and synchronization between caches and the global RPKI, is outside the scope of this document, and depends on things such as how often routers feed from the caches, how often the operator feels the global RPKI changes significantly, etc.

As inter-cache synchronization within an operator’s network does not impact global RPKI resources, an operator may choose to synchronize quite frequently.

To relieve routers of the load of performing certificate validation, cryptographic operations, etc., the RPKI-Router protocol, [RFC6810], does not provide object-based security to the router. I.e. the router can not validate the data cryptographically from a well-known trust anchor. The router trusts the cache to provide correct data and relies on transport based security for the data received from the cache. Therefore the authenticity and integrity of the data from the cache should be well protected, see Section 7 of [RFC6810].

As RPKI-based origin validation relies on the availability of RPKI data, operators SHOULD locate RPKI caches close to routers that require these data and services in order to minimize the impact of likely failures in local routing, intermediate devices, long circuits, etc. One should also consider trust boundaries, routing bootstrap reachability, etc.

For example, a router should bootstrap from a chache which is reachable with minimal reliance on other infrastructure such as DNS or routing protocols. If a router needs its BGP and/or IGP to converge for the router to reach a cache, once a cache is reachable, the router will then have to reevaluate prefixes already learned via BGP. Such configurations should be avoided if reasonably possible.

If insecure transports are used between an operator’s cache and their router(s), the Transport Security recommendations in [RFC6810] SHOULD be followed. In particular, operators MUST NOT use insecure transports between their routers and RPKI caches located in other Autonomous Systems.

For redundancy, a router should peer with more than one cache at the same time. Peering with two or more, at least one local and others remote, is recommended.

If an operator trusts upstreams to carry their traffic, they may also trust the RPKI data those upstreams cache, and SHOULD peer with caches made available to them by those upstreams. Note that this places an obligation on those upstreams to maintain fresh and reliable caches, and to make them available to their customers. And, as usual, the recipient SHOULD re-validate the data.

A transit provider or a network with peers SHOULD validate origins in announcements made by upstreams, down-streams, and peers. They still should trust the caches provided by their upstreams.

Before issuing a ROA for a super-block, an operator MUST ensure that all sub-allocations from that block which are announced by other ASs, e.g. customers, have correct ROAs in the RPKI. Otherwise, issuing a ROA for the super-block will cause the announcements of sub-allocations with no ROAs to be viewed as Invalid, see [RFC6811]. While waiting for all sub-allocatees to register ROAs, the owner of the super-block may use live BGP data to populate ROAs as a proxy, and then safely issue a ROA for the super-block.

Use of RPKI-based origin validation removes any need to originate more specifics into BGP to protect against mis-origination of a less specific prefix. Having a ROA for the covering prefix will protect it.

To aid translation of ROAs into efficient search algorithms in routers, ROAs should be as precise as possible, i.e. match prefixes as announced in BGP. E.g. software and operators SHOULD avoid use of excessive max length values in ROAs unless operationally necessary.

One advantage of minimal ROA length is that the forged origin attack does not work for sub-prefixes that are not covered by overly long max length. E.g. if, instead of 10.0.0.0/16-24, one issues 10.0.0.0/16 and 10.0.42.0/24, a forged origin attack can not succeed against 10.0.666.0/24. They must attack the whole /16, which is more likely to be noticed because of its size.

Therefore, ROA generation software MUST use the prefix length as the max length if the user does not specify a max length.

RFC EDITOR PLEASE REMOVE THIS PARAGRAPH: The above example does not use a standard documentation prefix as it needs a /16 so that a /24 can hole punch. As anything longer than a /24 is not globally routed, a /24 with a /25 (or whatever) hole would not be realistic and the ops reader would spend their energy on that anomaly instead of the example.

Operators should be conservative in use of max length in ROAs. E.g., if a prefix will have only a few sub-prefixes announced, multiple ROAs for the specific announcements should be used as opposed to one ROA with a long max length.

Operators owning prefix P should issue ROAs for all ASs which may announce P. If a prefix is legitimately announced by more than one AS, ROAs for all of the ASs SHOULD be issued so that all are considered Valid.

In an environment where private address space is announced in eBGP the operator may have private RPKI objects which cover these private spaces. This will require a trust anchor created and owned by that environment, see [I-D.ietf-sidr-ltamgmt].

Operators issuing ROAs may have customers which announce their own prefixes and ASs into global eBGP but who do not wish to go though the work to manage the relevant certificates and ROAs. Operators SHOULD offer to provision the RPKI data for these customers just as they provision many other things for them.

While an operator using RPKI data MAY choose any polling frequency they wish for ensuring they have a fresh RPKI cache. However, if they use RPKI data as an input to operational routing decisions, they SHOULD ensure local caches inside their AS are synchronized with each other at least every four to six hours.

Operators should use tools which warn them of any impending ROA or certificate expiry which could affect the validity of their own data. Ghostbuster Records, see [RFC6493], can be used to facilitate contact with upstream CAs to effect repair.

4. Within a Network

Origin validation need only be done by edge routers in a network, those which border other networks/ASs.

A validating router will use the result of origin validation to influence local policy within its network, see Section 5. In deployment this policy should fit into the AS’s existing policy, preferences, etc. This allows a network to incrementally deploy validation-capable border routers.

The operator should be aware that RPKI-based origin validation, as any other policy change, can cause traffic shifts in their network. And, as with normal policy shift practice, a prudent operator has tools and methods to predict, measure, modify, etc.

5. Routing Policy

Origin validation based on the RPKI marks a received announcement as having an origin which is Valid, NotFound, or Invalid, see [RFC6811]. How this is used in routing should be specified by the operator’s local policy.

Local policy using relative preference is suggested to manage the uncertainty associated with a system in early deployment, applying local policy to eliminate the threat of unreachability of prefixes due to ill-advised certification policies and/or incorrect certification data. E.g. until the community feels comfortable relying on RPKI data, routing on Invalid origin validity, though at a low preference, MAY occur.

Operators should be aware that accepting Invalid announcements, no matter how de-preffed, will often be the equivalent of treating them as fully Valid. Consider having a ROA for AS 42 for prefix 10.0.0.0/16-24. A BGP announcement for 10.0.666.0/24 from AS 666 would be Invalid. But if policy is not configured to discard it, then longest match forwarding will send packets toward AS 666 no matter the value of local preference.

As origin validation will be rolled out incrementally, coverage will be incomplete for a long time. Therefore, routing on NotFound validity state SHOULD be done for a long time. As the transition moves forward, the number of BGP announcements with validation state NotFound should decrease. Hence an operator’s policy should not be overly strict, and should prefer Valid announcements, attaching a lower preference to, but still using, NotFound announcements, and dropping or giving a very low preference to Invalid announcements. Merely de-preffing Invalids is ill-advised, see previous paragraph.

Some providers may choose to set Local-Preference based on the RPKI validation result. Other providers may not want the RPKI validation result to be more important than AS-path length -- these providers would need to map RPKI validation result to some BGP attribute that is evaluated in BGP’s path selection process after AS-path is evaluated. Routers implementing RPKI-based origin validation MUST provide such options to operators.

Local-Preference may be used to carry both the validity state of a prefix along with its traffic engineering (TE) characteristic(s). It is likely that an operator already using Local-Preference will have to change policy so they can encode these two separate characteristics in the same BGP attribute without negative impact or opening privilege escalation attacks. E.g. do not encode validation state in higher bits than used for TE.

When using a metric which is also influenced by other local policy, an operator should be careful not to create privilege upgrade vulnerabilities. E.g. if Local Pref is set depending on validity state, be careful that peer community signaling SHOULD NOT upgrade an Invalid announcement to Valid or better.

Announcements with Valid origins should be preferred over those with NotFound or Invalid origins, if Invalid origins are accepted at all.

Announcements with NotFound origins should be preferred over those with Invalid origins.

Announcements with Invalid origins SHOULD NOT be used, but may be used to meet special operational needs. In such circumstances, the announcement should have a lower preference than that given to Valid or NotFound.

When first deploying origin validation, it may be prudent to not drop announcements with Invalid orgins until inspection of logs, SNMP, or other data indicate that the correct result would be obtained.

Validity state signaling SHOULD NOT be accepted from a neighbor AS. The validity state of a received announcement has only local scope due to issues such as scope of trust, RPKI synchrony, and [I-D.ietf-sidr-ltamgmt].

6. Notes and Recommendations

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no ’fix’ for this, it is the nature of distributed data with distributed caches.

Operators should beware that RPKI caches are loosely synchronized, even within a single AS. Thus, changes to the validity state of prefixes could be different within an operator’s network. In addition, there is no guaranteed interval from when an RPKI cache is updated to when that new information may be pushed or pulled into a set of routers via this protocol. This may result in sudden shifts of traffic in the operator’s network, until all of the routers in the AS have reached equilibrium with the validity state of prefixes reflected in all of the RPKI caches.

It is hoped that testing and deployment will produce advice on relying party cache loading and timing.

There is some uncertainty about the origin AS of aggregates and what, if any, ROA can be used. The long range solution to this is the deprecation of AS-SETs, see [RFC6472].

As reliable access to the global RPKI and an operator’s caches (and possibly other hosts, e.g. DNS root servers) is important, an operator should take advantage of relying party tools which report changes in BGP or RPKI data which would negatively affect validation of such prefixes.

Operators should be aware that there is a trade-off in placement of an RPKI repository in address space for which the repository’s content is authoritative. On one hand, an operator will wish to maximize control over the repository. On the other hand, if there are reachability problems to the address space, changes in the repository to correct them may not be easily accessed by others.

Operators who manage certificates should associate RPKI Ghostbusters Records (see [RFC6493]) with each publication point they control. These are publication points holding the CRL, ROAs, and other signed objects issued by the operator, and made available to other ASs in support of routing on the public Internet.

Routers which perform RPKI-based origin validation must support Four-octet AS Numbers (see [RFC6793]), as, among other things, it is not reasonable to generate ROAs for AS 23456.

Software which produces filter lists or other control forms for routers where the target router does not support Four-octet AS Numbers (see [RFC6793]) must be prepared to accept Four-octet AS Numbers and generate the appropriate two-octet output.

As a router must evaluate certificates and ROAs which are time dependent, routers’ clocks MUST be correct to a tolerance of approximately an hour.

Servers should provide time service, such as [RFC5905], to client routers.

7. Security Considerations

As the BGP origin AS of an update is not signed, origin validation is open to malicious spoofing. Therefore, RPKI-based origin validation is expected to deal only with inadvertent mis-advertisement.

Origin validation does not address the problem of AS-Path validation. Therefore paths are open to manipulation, either malicious or accidental.

As BGP does not ensure that traffic will flow via the paths it advertises, the data plane may not follow the control plane.

Be aware of the class of privilege escalation issues discussed in Section 5 above.

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

The author wishes to thank Shane Amante, Rob Austein, Steve Bellovin, Jay Borkenhagen, Wes George, Seiichi Kawamura, Steve Kent, Pradosh Mohapatra, Chris Morrow, Sandy Murphy, Eric Osterweil, Keyur Patel, Heather and Jason Schiller, John Scudder, Kotikalapudi Sriram, Maureen Stillman, and Dave Ward.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

[RFC6490] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 6490, February 2012.

[RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, February 2012.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, December 2012.

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, January 2013.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, January 2013.

10.2. Informative References

[I-D.ietf-sidr-ltamgmt] Reynolds, M., Kent, S., and M. Lepinski, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-08 (work in progress), April 2013.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, February 2010.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6472] Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", BCP 172, RFC 6472, December 2011.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, February 2012.

[iab] , "IAB statement on the RPKI", , <http://www.iab.org/documents/correspondence-reports-documents/docs2010/iab-statement-on-the-rpki/>.

[rcynic] , "rcynic read-me", , <http://rpki.net/rcynic>.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

P. Mohapatra
Cisco Systems
J. Scudder
Juniper Networks
D. Ward
Cisco Systems
R. Bush
Internet Initiative Japan, Inc.
R. Austein
Dragon Research Labs
March 12, 2012

BGP Prefix Origin Validation
draft-ietf-sidr-pfx-validate-04

Abstract

To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

Status of this Memo

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Table of Contents

1. Introduction

A BGP route associates an address prefix with a set of autonomous systems (AS) that identify the interdomain path the prefix has traversed in the form of BGP announcements. This set is represented as the AS_PATH attribute in BGP [RFC4271] and starts with the AS that originated the prefix. To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

The Resource Public Key Infrastructure (RPKI) describes an approach to build a formally verifiable database of IP addresses and AS numbers as resources. The overall architecture of RPKI as defined in [RFC6480] consists of three main components:

- A public key infrastructure (PKI) with the necessary certificate objects,
- Digitally signed routing objects,
- A distributed repository system to hold the objects that would also support periodic retrieval.

The RPKI system is based on resource certificates that define extensions to X.509 to represent IP addresses and AS identifiers [RFC3779], thus the name RPKI. Route Origin Authorizations (ROA) [RFC6482] are separate digitally signed objects that define associations between ASes and IP address blocks. Finally the repository system is operated in a distributed fashion through the IANA, RIR hierarchy, and ISPs.

In order to benefit from the RPKI system, it is envisioned that relying parties either at AS or organization level obtain a local copy of the signed object collection, verify the signatures, and process them. The cache must also be refreshed periodically. The exact access mechanism used to retrieve the local cache is beyond the scope of this document.

Individual BGP speakers can utilize the processed data contained in the local cache to validate BGP announcements. The protocol details to retrieve the processed data from the local cache to the BGP speakers is beyond the scope of this document (refer to [I-D.ietf-sidr-rpki-rtr] for such a mechanism). This document proposes a means by which a BGP speaker can make use of the processed data in order to assign a "validity state" to each prefix in a received BGP UPDATE message.

Note that the complete path attestation against the AS_PATH attribute of a route is outside the scope of this document.

Although RPKI provides the context for this draft, it is equally possible to use any other database which is able to map prefixes to their authorized origin ASes. Each distinct database will have its own particular operational and security characteristics; such characteristics are beyond the scope of this document.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Prefix-to-AS Mapping Database

The BGP speaker loads validated objects from the cache into local storage. The objects loaded have the content (IP address, prefix length, maximum length, origin AS number). We refer to such a locally stored object colloquially as a "ROA" in the discussion below although we note that this is not a strictly accurate use of the term.

We define several terms in addition to "ROA". Where these terms are used, they are capitalized:

- Prefix: (IP address, prefix length), interpreted as is customary (see [RFC4632]).
- Route: Data derived from a received BGP UPDATE, as defined in [RFC4271], Section 1.1. The Route includes one Prefix and an AS_PATH; it may include other attributes to characterize the prefix.
- ROA Prefix: The Prefix from a ROA.
- ROA ASN: The origin AS number from a ROA.
- Route Prefix: The Prefix derived from a route.
- Route Origin ASN: The origin AS number derived from a Route. The origin AS number is the rightmost AS in the final segment of the

AS_PATH attribute in the Route if that segment is of type AS_SEQUENCE, or NONE if the final segment of the AS_PATH attribute is of any type other than AS_SEQUENCE. No ROA can match an origin AS number of "NONE". No Route can match a ROA whose origin AS number is zero.

- Covered: A Route Prefix is said to be Covered by a ROA when the ROA prefix length is less than or equal to the Route prefix length and the ROA prefix address matches the Route prefix address for all bits specified by the ROA prefix length. (This is simply a statement of the well-known concept of determining a prefix match.)

- Matched: A Route Prefix is said to be Matched by a ROA when the Route Prefix is Covered by that ROA and in addition, the Route prefix length is less than or equal to the ROA maximum length and the Route Origin ASN is equal to the ROA ASN, keeping in mind that a ROA ASN of zero can never be matched, nor can a route origin AS number of "NONE".

Given these definitions, any given BGP Route will be found to have one of the following "validation states":

- NotFound: No ROA Covers the Route Prefix.

- Valid: At least one ROA Matches the Route Prefix.

- Invalid: At least one ROA Covers the Route Prefix, but no ROA Matches it.

When a BGP speaker receives an UPDATE from one of its EBGP peers, it SHOULD perform a lookup as described above for each of the Routes in the UPDATE message. The "validation state" of the Route SHOULD be set to reflect the result of the lookup. Note that the validation state of the Route does not determine whether the Route is stored in the local BGP speaker’s Adj-RIB-In. This procedure SHOULD NOT be performed for Routes learned from peers of types other than EBGP. (Any of these MAY be overridden by configuration.) The suggested implementation should consider the "validation state" as described in the document as a local property or attribute of the Route. If validation is not performed on a Route, the implementation SHOULD initialize the validation state of such a route to "Valid".

Use of the validation state is discussed in Section 3 and Section 5.

We observe that a Route can be Matched or Covered by more than one ROA. This procedure does not mandate an order in which ROAs must be visited; however, the "validation state" output is fully determined.

2.1. Pseudo-Code

The following pseudo-code illustrates the procedure above. In case of ambiguity, the procedure above, rather than the pseudo-code, should be taken as authoritative.

```
//Input are the variables derived from a BGP UPDATE message
//that need to be validated.
//
//The input prefix is comprised of prefix.address and
//prefix.length.
//
//Collectively, the prefix and origin_as correspond to the
//Route defined in the preceding section.
input = {prefix, origin_as};

//Initialize result to "NotFound" state
result = BGP_PFXV_STATE_NOT_FOUND;

//pfx_validate_table organizes all the ROA entries retrieved
//from the RPKI cache based on the IP address and the prefix
//length field. There can be multiple such entries that match
//the input. Iterate through all of them.
entry = next_lookup_result(pfx_validate_table, input.prefix);

while (entry != NULL) {
  prefix_exists = TRUE;

  if (input.prefix.length <= entry->max_length) {
    if (input.origin_as != NONE
        && entry->origin_as != 0
        && input.origin_as == entry->origin_as) {
      result = BGP_PFXV_STATE_VALID;
      return (result);
    }
  }
  entry = next_lookup_result(pfx_validate_table, input.prefix);
}

//If pfx_validate_table contains one or more prefixes that
//match the input, but none of them resulted in a "valid"
//outcome since the origin_as did not match, return the
//result state as "invalid". Else the initialized state of
//"NotFound" applies to this validation operation.
if (prefix_exists == TRUE) {
  result = BGP_PFXV_STATE_INVALID;
}
```

```
   return (result);
```

3. Policy Control

An implementation MUST provide the ability to match and set the validation state of routes as part of its route policy filtering function. Use of validation state in route policy is elaborated in Section 5. For more details on operational policy considerations, see [I-D.ietf-sidr-origin-ops].

An implementation MUST support Four-Octet AS Numbers, [RFC4893].

4. Interaction with Local Cache

Each BGP speaker supporting prefix validation as described in this document is expected to communicate with one or more RPKI caches, each of which stores a local copy of the global RPKI database. The protocol mechanisms used to gather and validate these data and present them to BGP speakers are described in [I-D.ietf-sidr-rpki-rtr].

The prefix-to-AS mappings used by the BGP speaker are expected to be updated over time. When a mapping is added or deleted, the implementation MUST re-validate any affected prefixes. An "affected prefix" is any prefix that was matched by a deleted or updated mapping, or could be matched by an added mapping.

5. Deployment Considerations

Once a Route is selected for validation, it is categorized according the procedure given in Section 2. Subsequently, routing policy as discussed in Section 3 can be used to take action based on the validation state.

Policies which could be implemented include filtering routes based on validation state (for example, rejecting all "invalid" routes) or adjusting a route’s degree of preference in the selection algorithm based on its validation state. The latter could be accomplished by adjusting the value of such attributes as LOCAL_PREF. Considering invalid routes for BGP decision process is a pure local policy matter and should be done with utmost care.

In some cases (particularly when the selection algorithm is influenced by the adjustment of a route property that is not propagated into IBGP) it could be necessary for routing correctness to propagate the validation state to the IBGP peer. This can be accomplished on the sending side by setting a community or extended community based on the validation state, and on the receiving side by matching the (extended) community and setting the validation state.

6. Acknowledgments

The authors wish to thank Rex Fernando, Hannes Gredler, Mouhcine Guennoun, Russ Housley, Junaid Israr, Miya Kohno, Shin Miyakawa, Taka Mizuguchi, Hussein Mouftah, Keyur Patel, Tomoya Yoshida, and Kannan Varadhan. The authors are grateful for the feedback from the members of the SIDR working group.

Junaid Israr’s contribution to this specification was part of his PhD research work and thesis at University of Ottawa.

7. IANA Considerations

8. Security Considerations

Although this specification discusses one portion of a system to validate BGP routes, it should be noted that it relies on a database (RPKI or other) to provide validation information. As such, the security properties of that database must be considered in order to determine the security provided by the overall solution. If "invalid" routes are blocked as this specification suggests, the overall system provides a possible denial-of-service vector, for example if an attacker is able to inject or remove one or more records in the validation database, it could lead an otherwise valid route to be marked as invalid.

In addition, this system is only able to provide limited protection against a determined attacker -- the attacker need only prepend the "valid" source AS to a forged BGP route announcement in order to defeat the protection provided by this system.

This mechanism does not protect against "AS in the middle attacks" or provide any path validation. It only attempts to verify the origin. In general, this system should be thought of more as a protection against misconfiguration than as true "security" in the strong sense.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

9.2. Informational References

[I-D.ietf-sidr-origin-ops] Bush, R., "RPKI-Based Origin Validation Operation", draft-ietf-sidr-origin-ops-15 (work in progress), March 2012.

[I-D.ietf-sidr-rpki-rtr] Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-26 (work in progress), February 2012.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

Authors’ Addresses

Pradosh Mohapatra
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: pmohapat@cisco.com

John Scudder
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA

Email: jgs@juniper.net

David Ward
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dward@cisco.com

Randy Bush
Internet Initiative Japan, Inc.
5147 Crystal Springs
Bainbridge Island, Washington 98110
USA

Email: randy@psg.com

Rob Austein
Dragon Research Labs

Email: sra@hactrn.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 02, 2013

P. Mohapatra
Cisco Systems
J. Scudder
Juniper Networks
D. Ward
Cisco Systems
R. Bush
Internet Initiative Japan
R. Austein
Dragon Research Labs
October 2012

BGP Prefix Origin Validation
draft-ietf-sidr-pfx-validate-10

Abstract

To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

Status of This Memo

This Internet-Draft will expire on April 02, 2013.

Copyright Notice

Table of Contents

1. Introduction

A BGP route associates an address prefix with a set of autonomous systems (AS) that identify the interdomain path the prefix has traversed in the form of BGP announcements. This set is represented as the AS_PATH attribute in BGP [RFC4271] and starts with the AS that originated the prefix. To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

The Resource Public Key Infrastructure (RPKI) describes an approach to build a formally verifiable database of IP addresses and AS numbers as resources. The overall architecture of RPKI as defined in [RFC6480] consists of three main components:

- A public key infrastructure (PKI) with the necessary certificate objects,

- Digitally signed routing objects,

- A distributed repository system to hold the objects that would also support periodic retrieval.

The RPKI system is based on resource certificates that define extensions to X.509 to represent IP addresses and AS identifiers [RFC3779], thus the name RPKI. Route Origin Authorizations (ROA) [RFC6482] are separate digitally signed objects that define associations between ASes and IP address blocks. Finally the repository system is operated in a distributed fashion through the IANA, RIR hierarchy, and ISPs.

In order to benefit from the RPKI system, it is envisioned that relying parties either at AS or organization level obtain a local copy of the signed object collection, verify the signatures, and process them. The cache must also be refreshed periodically. The exact access mechanism used to retrieve the local cache is beyond the scope of this document.

Individual BGP speakers can utilize the processed data contained in the local cache to validate BGP announcements. The protocol details to retrieve the processed data from the local cache to the BGP speakers is beyond the scope of this document (refer to [I-D.ietf-sidr-rpki-rtr] for such a mechanism). This document proposes a means by which a BGP speaker can make use of the processed data in order to assign a "validation state" to each prefix in a received BGP UPDATE message.

Note that the complete path attestation against the AS_PATH attribute of a route is outside the scope of this document.

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no ’fix’ for this, it is the nature of distributed data with distributed caches.

Although RPKI provides the context for this draft, it is equally possible to use any other database which is able to map prefixes to their authorized origin ASes. Each distinct database will have its own particular operational and security characteristics; such characteristics are beyond the scope of this document.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without any normative meaning.

2. Prefix-to-AS Mapping Database

The BGP speaker loads validated objects from the cache into local storage. The objects loaded have the content (IP address, prefix length, maximum length, origin AS number). We refer to such a locally stored object as a "Validated ROA Payload" or "VRP".

We define several terms in addition to "VRP". Where these terms are used, they are capitalized:

- Prefix: (IP address, prefix length), interpreted as is customary (see [RFC4632]).
- Route: Data derived from a received BGP UPDATE, as defined in [RFC4271], Section 1.1. The Route includes one Prefix and an AS_PATH; it may include other attributes to characterize the prefix.
- VRP Prefix: The Prefix from a VRP.
- VRP ASN: The origin AS number from a VRP.
- Route Prefix: The Prefix derived from a route.

- Route Origin ASN: The origin AS number derived from a Route as follows:
  - the rightmost AS in the final segment of the AS_PATH attribute in the Route if that segment is of type AS_SEQUENCE, or
  - the BGP speaker’s own AS number if that segment is of type AS_CONFED_SEQUENCE or AS_CONFED_SET or if the AS_PATH is empty, or
  - the distinguished value "NONE" if the final segment of the AS_PATH attribute is of any other type.
- Covered: A Route Prefix is said to be Covered by a VRP when the VRP prefix length is less than or equal to the Route prefix length, and the VRP prefix address and the Route prefix address are identical for all bits specified by the VRP prefix length.(I.e. the Route prefix is either identical to the VRP prefix or a more specific of the VRP prefix.)
- Matched: A Route Prefix is said to be Matched by a VRP when the Route Prefix is Covered by that VRP and in addition, the Route prefix length is less than or equal to the VRP maximum length and the Route Origin ASN is equal to the VRP ASN.

Given these definitions, any given BGP Route will be found to have one of the following "validation states":

- NotFound: No VRP Covers the Route Prefix.
- Valid: At least one VRP Matches the Route Prefix.
- Invalid: At least one VRP Covers the Route Prefix, but no VRP Matches it.

We observe that no VRP can have the value "NONE" as its VRP ASN. Thus a Route whose Origin ASN is "NONE" cannot be Matched by any VRP. Similarly, no valid Route can have an Origin ASN of zero [I-D.ietf-idr-as0]. Thus no Route can be Matched by a VRP whose ASN is zero.

When a BGP speaker receives an UPDATE from a neighbor, it SHOULD perform a lookup as described above for each of the Routes in the UPDATE message. The lookup SHOULD also be applied to routes which are redistributed into BGP from another source, such as another protocol or a locally defined static route. An implementation MAY provide configuration options to control which routes the lookup is applied to. The "validation state" of the Route MUST be set to reflect the result of the lookup. The implementation should consider the "validation state" as described in the document as a local property or attribute of the Route. If validation is not performed on a Route, the implementation SHOULD initialize the "validation state" of such a route to "NotFound".

Use of the validation state is discussed in Section 3 and Section 5. An implementation MUST NOT exclude a route from the Adj-RIB-In or from consideration in the decision process as a side-effect of its validation state, unless explicitly configured to do so.

We observe that a Route can be Matched or Covered by more than one VRP. This procedure does not mandate an order in which VRPs must be visited; however, the "validation state" output is fully determined.

2.1. Pseudo-Code

The following pseudo-code illustrates the procedure above. In case of ambiguity, the procedure above, rather than the pseudo-code, should be taken as authoritative.

```
result = BGP_PFXV_STATE_NOT_FOUND;

//Iterate through all the Covering entries in the local VRP
//database, pfx_validate_table.
entry = next_lookup_result(pfx_validate_table, route_prefix);

while (entry != NULL) {
  prefix_exists = TRUE;

  if (route_prefix_length <= entry->max_length) {
    if (route_origin_as != NONE
        && entry->origin_as != 0
        && route_origin_as == entry->origin_as) {
      result = BGP_PFXV_STATE_VALID;
      return (result);
    }
  }
  entry = next_lookup_result(pfx_validate_table, input.prefix);
}

//If one or more VRP entries Covered the route prefix, but
//no one Matched, return "Invalid" validation state.
if (prefix_exists == TRUE) {
  result = BGP_PFXV_STATE_INVALID;
}

return (result);
```

3. Policy Control

An implementation MUST provide the ability to match and set the validation state of routes as part of its route policy filtering function. Use of validation state in route policy is elaborated in

Section 5. For more details on operational policy considerations, see [I-D.ietf-sidr-origin-ops].

An implementation MUST also support Four-Octet AS Numbers, [RFC4893].

4. Interaction with Local Cache

Each BGP speaker supporting prefix validation as described in this document is expected to communicate with one or more RPKI caches, each of which stores a local copy of the global RPKI database. The protocol mechanisms used to gather and validate these data and present them to BGP speakers are described in [I-D.ietf-sidr-rpki-rtr].

The prefix-to-AS mappings used by the BGP speaker are expected to be updated over time. When a mapping is added or deleted, the implementation MUST re-validate any affected prefixes and run the BGP decision process if needed. An "affected prefix" is any prefix that was matched by a deleted or updated mapping, or could be matched by an added or updated mapping.

5. Deployment Considerations

Once a Route is selected for validation, it is categorized according the procedure given in Section 2. Subsequently, routing policy as discussed in Section 3 can be used to take action based on the validation state.

Policies which could be implemented include filtering routes based on validation state (for example, rejecting all "invalid" routes) or adjusting a route’s degree of preference in the selection algorithm based on its validation state. The latter could be accomplished by adjusting the value of such attributes as LOCAL_PREF. Considering invalid routes for BGP decision process is a pure local policy matter and should be done with utmost care.

In some cases (particularly when the selection algorithm is influenced by the adjustment of a route property that is not propagated into IBGP) it could be necessary for routing correctness to propagate the validation state to the IBGP peer. This can be accomplished on the sending side by setting a community or extended community based on the validation state, and on the receiving side by matching the (extended) community and setting the validation state.

6. Acknowledgments

The authors wish to thank Rex Fernando, Hannes Gredler, Mouhcine Guennoun, Russ Housley, Junaid Israr, Miya Kohno, Shin Miyakawa, Taka Mizuguchi, Hussein Mouftah, Keyur Patel, Tomoya Yoshida, Kannan Varadhan, Wes George, Jay Borkenhagen, and Sandra Murphy. The authors are grateful for the feedback from the members of the SIDR working group.

Junaid Israr’s contribution to this specification was part of his PhD research work and thesis at University of Ottawa.

7. IANA Considerations

[Note to RFC Editor: This section may be removed on publication]

This document has no IANA considerations.

8. Security Considerations

Although this specification discusses one portion of a system to validate BGP routes, it should be noted that it relies on a database (RPKI or other) to provide validation information. As such, the security properties of that database must be considered in order to determine the security provided by the overall solution. If "invalid" routes are blocked as this specification suggests, the overall system provides a possible denial-of-service vector, for example if an attacker is able to inject or remove one or more records in the validation database, it could lead an otherwise valid route to be marked as invalid.

In addition, this system is only able to provide limited protection against a determined attacker -- the attacker need only prepend the "valid" source AS to a forged BGP route announcement in order to defeat the protection provided by this system.

This mechanism does not protect against "AS in the middle attacks" or provide any path validation. It only attempts to verify the origin. In general, this system should be thought of more as a protection against misconfiguration than as true "security" in the strong sense.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3779] Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[RFC4271] Rekhter, Y., Li, T. and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[RFC6482] Lepinski, M., Kent, S. and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

9.2. Informational References

[I-D.ietf-idr-as0] Kumari, W., Bush, R., Schiller, H. and K. Patel, "Codification of AS 0 processing.", Internet-Draft draft-ietf-idr-as0-06, August 2012.

[I-D.ietf-sidr-origin-ops] Bush, R., "RPKI-Based Origin Validation Operation", Internet-Draft draft-ietf-sidr-origin-ops-19, August 2012.

[I-D.ietf-sidr-rpki-rtr] Bush, R. and R. Austein, "The RPKI/Router Protocol", Internet-Draft draft-ietf-sidr-rpki-rtr-26, February 2012.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

Authors’ Addresses

Pradosh Mohapatra
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: pmohapat@cisco.com

John Scudder
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA

Email: jgs@juniper.net

David Ward
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dward@cisco.com

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA 98110
USA

Email: randy@psg.com

Rob Austein
Dragon Research Labs

Email: sra@hactrn.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 4, 2012

R. Gagliano
K. Patel
B. Weis
Cisco Systems

March 5, 2012

BGPSEC router key roll-over as an alternative to beaconing
draft-rogaglia-sidr-bgpsec-rollover-00

Abstract

The current BGPSEC draft documents do not specifies a key roll-over process for routers. This document describes a possible key roll-over process and explores its impact to mitigate replay attacks and eliminate the need for beaconing in BGPSEC.

Status of this Memo

This Internet-Draft will expire on July 4, 2012.

Copyright Notice

Table of Contents

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

In BGPSEC, a key roll-over (or re-keying) is the process of changing the router’s key pair, issuing the correspondent new End-Entity certificates and revoke the old one. This process will need to happen at regular intervals normally due to local policies at each network.

During a roll-over process, a router needs to generate BGP UPDATE messages in order to signal the new key to be used to its neighbors. So, intuitively, a frequent key roll-over process has similar effects as the beaconing process proposed by the BGPSEC base documents to protect a BGPSEC attribute against a re-play attack. However, there are a number of operational details to be considered if the expire time field in the BGPSEC attribute is removed.

This document details a possible key roll-over process in BGPSEC and explores the operational environment where key roll-overs could be used as a protection against a re-play attach against BGPSEC

3. Key Roll-over in BGPSEC

The key roll-over process in BGPSEC has not been well defined yet. However, this will be a mandatory process due to some of the following causes:

BGPSEC scheduled roll-over: BGPSEC certificates have an expiration date (NotValidAfter). Although it is possible to generate a new certificate without changing the key pair, it is normally good practice to adopt the policy of using a new key pair in every roll-over event.

BGPSEC certificate fields changes: A BGPSEC certificate field’s information (such as the ASN or the Subject) may need to be changed. The normal process requires the roll-over of the old certificate with a new key pair and the revocation of the old certificate.

BGPSEC emergency roll-over Some special circumstances (such as a compromised key) may require the roll-over of a BGPSEC certificate.

It should be clear at this point that a key roll-over process is required for BGPSEC. The next section describes how this process may be implemented.

3.1. A proposed process for BGPSEC key roll-over

The BGPSEC key roll-over process should be very tighten to the key provisioning mechanisms that would be in place. The key provisioning mechanisms for BGPSEC are not yet documented. We will assume that such an automatic provisioning mechanism will be in place (a possible provisioning mechanism when the private key lives only inside the BGP speaker is the Enrollment over Secure Transport (EST). This protocol will allow BGPSEC code to include automatic re-keying scripts with minimum development cost.

When the same private key is shared by different routers, a mechanism to distribute the private key will need to be implemented. A possible solution may include the transmission of the private key over a secure channel. The PKIX WG has started work on this sense by adopting [I-D.ietf-pkix-cmc-serverkeygeneration]

If we work under the assumption that an automatic mechanism will exist to rollover a BGPSEC certificate, a possible process could be:

1. New Certificate Pre-Publication: The first step in the rollover mechanism is to pre-publish the new public key. In order to

accomplish this goal, the new key pair and certificate will need to be generated and published on the correspondent RPKI repository. This process will vary in every environment as it will depend on where the keys are located (either in every router or on a centralized server), if the RPKI CA is hosted at the ISP or at an external party (i.e. needs to use the RPKI provisioning protocol) and finally if the repository is also local or hosted (i.e. will need to use the RPKI-Repository protocol.)

2. Stage Period: A stage period will be required from the time a new certificate is published in the RPKI global repository until the time it is fetched by RPKI caches around the globe. The exact minimum staging time is not clear and will require experimental results from RPKI. Design documents mention a lower limit of 24 hours. If rollovers will be done frequently and we want to avoid the stage period in case of emergency rollover needs, an administrator can always provision two certificate for every router. In this case when the rollover operation is needed, the cache servers around the globe would already have the new keys.

3. Twilight: At this moment, the BGP speaker that uses the key been rolled-over will stop using the OLD key for signing and start using the NEW key. Also, the router will generate appropriate BGP UPDATES just as in the typical operation of refreshing outbound BGP polices.

4. CRL Publication: As part of the rollover process, a CA MAY decide that it will publish the serial number of the OLD BGPSEC certificate on its CRL. It may also be the case that the CA will just let the certificate to expire and not update its CRL.

5. RPKI-Router Protocol Withdrawal: Either due to the inclusion of the OLD certificate serial number or the expiration of the certificate’s validation, the RPKI cache servers around the globe will need to communicate to its RTR peers that the OLD certificate’s public key is not longer valid (withdrawal message). It is not documented yet what will be a router’s reaction to a RTR withdrawal message but it should include the removal of any RIB entry that includes a BGPSEC attribute signed with that key and the generation of the correspondent BGP WITHDRAWS (either implicit or explicit).

To conclude this section, we can say that the proposed rollover mechanism will depend on the existence of an automatic provisioning process for BGPSEC certificates, that it will required a staging mechanism given by RPKI propagation time of around 24hours and that it will generate BGP UPDATES for all prefixes in the router been re-keying.

4. BGPSEC key rollover as a measure against replays attacks in BGPSEC

There are two typical measures to mitigate replay attacks: addition of a timestamp or addition of a serial number. Currently BGPSEC offers a timestamp (expiration time) as a protection against re-play attacks of BGPSEC messages. The process requires all BGP Speakers that originate a BGP UPDATE to beaconing the message before its expiration time. This requirement changes a long standing BGP operation practice and the community have been searching for alternatives.

4.1. BGPSEC beaconing challenges

To be completed

4.2. BGPSEC Re-play attack window requirement

The BGPSEC Ops document give some ideas of requirements for the re-play attack in BGPSEC. For the vast majority of the prefixes, the requirement will be in the order of days or weeks. For a very small fraction, but critical, of the prefixes, the requirement may be in the order of hours.

4.3. BGPSEC key rollover as a mechanism to protect against replay attacks

The question we would like to ask is: can key rollover provide us a similar protection against re-play attacks without the need for beaconing?

The answer is that YES when the window requirement is in the order of days and the router re-keying is the edge router of the origin AS. By using re-keying, you are letting the BGPSEC certificate validation time as your timestamp against replay attacks. However, the use of frequent key rollovers comes with an additional administrative cost and risks if the process fails. As documented before, re-keying should be supported by automatic tools and for the great majority of the Internet it will be done with good lead time to correct any inconvenient in the process.

For a transit AS that also originates its BGP UPDATES for its own prefixes, the key rollover process may generate a large number of UPDATE messages (even the complete DFZ). For this reason, it is recommended that routers in this scenario been provisioned with two certificates: one to sign BGP UPDATES in transit and a second one to sign BGP UPDATE for prefixes originated in its AS. Only the second certificate should be frequently rolled-over.

Advantage of Re-keying as re-play attack protection mechanism:

1. Does not require beaconing
2. All timestamps policies are maintained in RPKI
3. Additional administrative cost is paid by the provider that wants to protect its infrastructure
4. Can be implemented in coordination with planned topology changes by either origin ASes or transit ASes (if I am changing providers, I rollover)
5. Eliminates the discussion on who has the authority over the expiration time

Disadvantage of Re-keying as re-play attack protection mechanism:

1. More administrative load due to frequent rollover, although how frequent is still not clear.
2. Minimum window size bounded by RPKI propagation time to RPKI caches. If pre-provisioning done ahead of time, it means 24 hours minimum in paper. However, more experimentation is needed when RPKI and cache servers are more massively deployed.
3. Increases dynamic of RPKI repository
4. More load on RPKI caches, but they are meant to do this work.

5. IANA Considerations

No IANA considerations

6. Security Considerations

No security considerations.

7. Acknowledgements

None yet

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.

8.2. Informative References

[I-D.ietf-pkix-cmc-serverkeygeneration]
Schaad, J., Timmel, P., and S. Turner, "CMC Extensions: Server Key Generation", draft-ietf-pkix-cmc-serverkeygeneration-00 (work in progress), January 2012.

[I-D.ietf-sidr-origin-validation-signaling]
Mohapatra, P., Patel, K., Scudder, J., Ward, D., and R. Bush, "BGP Prefix Origin Validation State Extended Community", draft-ietf-sidr-origin-validation-signaling-00 (work in progress), November 2010.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-01 (work in progress), February 2011.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors’ Addresses

Roque Gagliano
Cisco Systems
Avenue des Uttins 5
Rolle, VD 1180
Switzerland

Email: rogaglia@cisco.com

Keyur Patel
Cisco Systems
170 W. Tasman Driv
San Jose, CA 95134
CA

Email: keyupate@cisco.com

Brian Weis
Cisco Systems
170 W. Tasman Driv
San Jose, CA 95134
CA

Email: bew@cisco.com

Network Working Group
Internet-Draft
Intended status: BCP
Expires: September 2, 2012

S. Turner
IECA, Inc.
K. Patel
Cisco Systems
R. Bush
Internet Initiative Japan, Inc.

March 5, 2012

Router Keying for BGPsec
draft-ymbk-bgpsec-rtr-rekeying-00

Abstract

BGPsec-speaking routers must be provisioned with private keys and the corresponding public key must be published in the global Resource PKI. This document describes two ways of doing so, router-driven and operator-driven.

Status of this Memo

This Internet-Draft will expire on September 6, 2012.

Copyright Notice

Table of Contents

Table of Contents

1. Introduction

BGPsec-speaking routers must be provisioned with private keys and the corresponding public key must be published in the global RPKI (Resource Public Key Infrastructure). Note that the public key is published in the RPKI in the form of a certificate [I-D.sidr-bgpsec-pki-profiles]. This document describes two methods for generating the necessary public/private key-pair: router-driven and operator-driven.

In the router-driven method, the router generates its own public/private key-pair, uses the private key to sign a certification request [I-D.sidr-bgpsec-pki-profiles] (a PKCS#10 - includes the public key), and sends the certification request to the RPKI CA (Certification Authority). The CA returns a PKCS#7, which includes the certified public key in the form of a certificate, to the router and the CA also publishes the certificate in the RPKI.

The router-driven model mirrors the model used by most PKI subscribers. In many cases, the private key never leaves trusted storage (e.g., HSM (Hardware Security Model)). This is by design and supports CPs (Certification Policies), often times for human subscribers, that require the private key only ever be controlled by the subscriber to ensure that no one can impersonate the subscriber.

For non-humans, this model does not always work. For example, when an operator wants to support hot-swappable routers the same private key needs to be installed in the soon-to-be online router that was installed in the soon-to-be offline router. This motivated the operator-driven model.

In the operator-driven model, the operator generates the private/public key-pair and sends them to the router in a PKCS#8 [RFC5958].

In both cases, the key pair is for algorithms defined in [I-D.sidr-bgpsec-algs]. The first version specifies ECDSA on the P-256 curve.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

It is assumed that the reader understands BGPsec, see [I-D.lepinski-bgpsec-overview] and [I-D.lepinski-bgpsec-protocol], and the RPKI, see [RFC6480] and [I-D.sidr-bgpsec-pki-profiles].

3. Router-Generated Keys

For router-generated keys, the public/private keys are made by the router, a PKCS#10 is made by the router, the PKCS#10 is signed by the private key. The CA returns a PKCS#7 and the router picks the certificate out of the PKCS#7. Even if the operator can not get the private key off the router this still provides a linkage between a private key and a router.

4. Operator-Generated Keys

For operator-generated keys, the public/private keys are made by the operator with their RPKI management software. The private key pair MUST be as specified in [RFC5915], which supports ECDSA keys. That format MUST then be inserted to a PKCS#8 [RFC5958] along with the certificate. If the operator wants to ship the keys around they can use the .p8 file extension and optional PEM encoding also from [RFC5958].

EDITOR NOTE: One thing we should consider is whether the certificate needs to returned to the router like in the router-generated keys method. PKCS#8 supports including the certificate so it’s not a big deal to add it if we do.

5. Provisioning a New Router

When commissioning a new router, the operator may use either of the above methods.

Using the Router-Generated Keys method, see Section 3, the operator decides on the AS number and the BGP RouterID of the router, logs on to the new router using the craft port, ssh, etc., and requests that the router generate a public/private key-pair and generate and sign (with the private key) a PKCS#10 request. The operator then off-loads the PKCS#10 request and uploads the request to their RPKI software management tools. The tools create and publish the RPKI Router-Key object for the public key, and return the PKCS#7. The operator uploads the PKCS#7 to the router which then extracts its certificate.

Using the Operator-Generated Key method, see Section 4, the operator decides on the AS number and the BGP RouterID of the new router and uses their RPKI software management tools to generate the public/private key-pair and publish the public key in the RPKI. The tools also produce the PKCS#8 object which the operator then uploads into the new router via the craft port, ssh, NetConf, etc. The router installs the PKS#8 and installs the public/private key-pair.</t>

6. Other Use Cases

Current router code generates private keys for uses such as ssh, but the private keys may not be seen or off-loaded via CLI or any other means. While this is good security, it creates difficulties when a routing engine or whole router must be replaced in the field and all software which accesses the router must be updated with the new keys. Also, the initial contact with a new routing engine requires trust in the public key presented on first contact.

To allow operators to quickly replace routers without requiring update and distribution of the corresponding public keys in the RPKI, routers SHOULD allow the private BGPsec key to be off-loaded via the CLI, NetConf (see [RFC6470]), SNMP, etc. This lets the operator upload the old private key via the mechanism used for Operator-Generated Keys, see Section 5.

7. Security Considerations

Keys could be intercepted in transport and the recipient, RPKI or router, would have no way of knowing a substitution had been made by a monkey in the middle. Hence transport security is strongly advised.

8. IANA Considerations

This document has no IANA Considerations.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5915] Turner, S. and D. Brown, "Elliptic Curve Private Key Structure", RFC 5915, June 2010.

[RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, August 2010.

9.2. Informative References

[RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", RFC 6470, February 2012.

[I-D.sidr-bgpsec-overview] Lepinski, M. and S. Turner, "An Overview of BGPSEC", draft-ietf-sidr-bgpsec-overview-01 (work in progress), October 2011.

[I-D.sidr-bgpsec-protocol] Lepinski, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol-01 (work in progress), October 2011.

[I-D.sidr-bgpsec-pki-profiles] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bgpsec-pki-profiles-01 (work in progress), December 2011.

[I-D.sidr-bgpsec-algs]

Turner, S., "BGP Algorithms, Key Formats, & Signature Formats", draft-ietf-sidr-bgpsec-algs-01 (work in progress), December 2011.

Appendix A. Examples

The examples provided in this appendix were generated using OpenSSL 0.9.8.r.

Appendix A.1. Operator-Generated Keys

To generate the EC public and private keys:

```
openssl ecparam -genkey -name secp256v1 -noout -out ecKey.pem
```

The result is (note this ought not be reproducible because each key better be unique, but you ought to get the same format):

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIEzFLfqklXUpodvaqGuivapVRzRxiITh4UdlJ/JTAgKxoAoGCCqGSM49
AwEHoUQDQgAEM4VgV/qUB06BZ9bzqYyXIfacC5NDr9yavwxfbZnGejIaeXXt2OO/
qkmQQq3E7m/GEJ+XFyciLv2da9waZMTVQg==
-----END EC PRIVATE KEY-----
```

To convert the result to PKCS#8, issue the following command:

```
openssl pkcs8 -topk8 -inform PEM -outform PEM -in ecKey.pem -out
ecKey-p8.pem -nocrypt
```

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgTMUt+qSVdSmh29qo
a6K9qlVHNHGIhOHhR2Un8lMCArGhRANCAAQzhWBX+pQHToFn1vOpjJch9pwLk0Ov
3Jq/DF9tmcZ6Mhp5de3Y47+qSZBCrcTub8YQn5cXJyIu/Z1r3BpkxNVC
-----END PRIVATE KEY-----
```

Appendix A.1. Router-Generated Keys

TBD

Authors' Addresses

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, Virginia 22031
US

Email: turners@ieca.com

Keyur Patel
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
US

Email: keyupate@cisco.com

Randy Bush
Internet Initiative Japan, Inc.
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com